



UNIVERSIDAD DE MÁLAGA



INGENIERÍA DEL SOFTWARE

DESARROLLO DE UNA APLICACIÓN ANDROID
PARA EL CONTROL DE CONTADORES

DEVELOPMENT OF AN ANDROID APPLICATION
FOR METER CONTROL

Realizado por
JOSÉ ANTONIO ESCOBAR BUENO

Tutorizado por
LUIS MANUEL LLOPIS TORRES

Departamento
LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN

MÁLAGA, (Febrero 2021)



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERÍA DEL SOFTWARE

DESARROLLO DE UNA APLICACIÓN ANDROID PARA EL CONTROL DE CONTADORES

DEVELOPMENT OF AN ANDROID APPLICATION FOR METER CONTROL

Realizado por
José Antonio Escobar Bueno

Tutorizado por
Luis Manuel Llopis Torres

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, FEBRERO DE 2021

Fecha defensa: Febrero de 2021



UNIVERSIDAD
DE MÁLAGA



Resumen

En los últimos años se están desarrollando multitud de aplicaciones para el Internet de las Cosas (Internet of Things). El concepto es añadir a las “cosas” capacidad de dar información en tiempo real de los datos que puede recoger. Uno de los campos de aplicación es el Smart metering, que es la capacidad para poder tener conocimiento en tiempo real de los datos de consumo, ya sea de agua, luz, o gas. Este conocimiento en tiempo real permite que los usuarios puedan auto regularse en el consumo y, por tanto, ahorrar en el recibo. Una solución sería la incorporación de una aplicación móvil para conocer los datos sobre la factura en tiempo real. El contexto de dicha aplicación será aplicado al consumo de agua en los regadíos de la provincia de Málaga y tendrá dos usuarios finales, los consumidores y los revisores, que podrán ir emitiendo informes a sus responsables.

Este trabajo fin de grado desarrolla dicha aplicación que, además, tiene otras funcionalidades como:

Recibir avisos sobre si va a haber algún corte de agua, luz o gas.

Saber el consumo y el total a pagar y poder llevar cierto control.

Poder emitir algún informe sobre incidencias.

Palabras clave:

Android, Aplicación android, Firebase, Android Studio, monitorización, MagicDraw

Abstract

In recent years, many applications for the Internet of Things are being developed. The concept is to add to the "things" ability to give real-time information on the data that can be collected. One of the fields of application is Smart metering, which is the ability to have real-time knowledge of consumption data, whether water, electricity, or gas. This real-time knowledge allows users to self-regulate consumption and, therefore, save on receipt. One solution would be the incorporation of a mobile application to know the data on the invoice in real time. The context of this application will be applied to water consumption in irrigation in the province of Malaga and will have two end users, consumers and reviewers, who may be issuing reports to their managers.

This final project will develop this application, which will also have other features such as:
Receive notices about whether there will be any water, electricity or gas outages.

Know the consumption and the total to pay and be able to keep some control.

Be able to issue a report on incidents.

Keywords:

Android, android App, Firebase, Android Studio, Monitoring, MagicDraw

Índice

Resumen	1
Abstract	1
Índice	1
Índice de tablas	1
Índice de figuras	1
Introducción	1
1.1 Motivación	1
1.2 Objetivos	2
1.3 Estructura de la memoria	3
1. Introducción	3
2. Sistemas Operativos Móviles.....	3
3. Tecnologías usadas	3
4. Librerías usadas	3
5. Análisis	3
6. Metodología de desarrollo	3
7. Bases de datos	4
8. Diseño de la aplicación	4
9. Desarrollo de la aplicación.....	4
10. Seguridad.....	4
11. Pruebas	4
12. Conclusiones.....	4
Apéndice A Manual de instalación	4
Apéndice B: Manual de usuario.....	4
Sistemas Operativos Móviles	1
2.1 Android	1
2.1.1 Versiones	1
2.1.2 Estructura	2
2.2 IOS	2
2.1.1 Versiones	2
2.1.2 Estructura	3
Tecnologías utilizadas.....	5
3.1 Java	5
3.2 MagicDraw	5
3.3 Firebase.....	5
3.4 Android Studio.....	6
3.5 Gradle	6
3.6 Draw.io.....	6
3.7 Trello.....	6

3.8 Git.....	6
3.9 Bitbucket.....	7
Librerías usadas	9
4.1 Material Design Icons.....	9
4.2 Apache POI.....	9
4.3 MPAndroidChart.....	9
4.4 Firebase.....	10
4.4.1 Firebase Authentication	10
4.4.2 Firebase Cloud Firestore.....	10
4.4.3 Firebase Cloud Storage	10
4.4.4 Firebase Crashlytics	10
4.4.4 Firebase Analytics	10
Análisis.....	11
5.1 Descripción.....	11
5.2 Obtención de requisitos	11
5.3 Requisitos funcionales	12
5.4 Requisitos no funcionales	14
5.5 Casos de uso.....	14
5.5.1 Diagramas de casos de uso.....	14
5.5.2 Lista de casos de uso	16
Metodología de desarrollo.....	49
6.1 Metodología.....	49
6.2 Entorno	50
6.3 Proceso	50
Bases de datos.....	53
7.1 SQL.....	53
7.2 No SQL	53
7.3 Base de datos de la aplicación	54
Diseño de la aplicación.....	57
8.1 Diseño de la base de datos	57
8.1.1 Diagramas de la base de datos	59
8.2 Diseño de la interfaz	60
8.3 Diseño del mockup	61
Desarrollo de la aplicación	63
9.1 Diagrama de clases	63
9.3 Estructura del proyecto.....	64
Seguridad	67
10.1 Seguridad de la app.....	67
10.2 Firebase Rules.....	68
10.3 Almacenamiento interno	68
Pruebas	69
11.1 Descripción.....	69
11.2 Emulador.....	69
11.3 ADB	70
11.4 Espresso	70
11.4.1 Pruebas de unidad	71
11.4.2 Pruebas de Integración.....	71

11.4.3 Pruebas de Extremo a extremo	71
11.5 Firebase Crashlytics	71
11.6 Firebase Test Lab.....	71
11.7 Pruebas realizadas	73
Conclusiones	83
12.1 Conclusiones	83
12.2 Líneas futuras	83
Bibliografía.....	85
Manual de Instalación	87
A.1 Requisitos	87
A.2 Instalación de la aplicación	87
A.2.1 Mediante la APK	87
A.2.2 Mediante el proyecto	89
A.3 Instalación de la base de datos.....	90
Manual de Usuario	93
B.1 Introducción	93
B.2 Usuario.....	95
B.2.1 Registro del usuario.....	95
B.2.2 Agregar un contador	95
B.2.3 Ver los datos de un contador	96
B.2.4 Eliminar un contador	97
B.2.5 Notificaciones de la aplicación	97
B.3 Administrador	97
B.3.1 Registro del administrador.....	98
B.3.2 Crear una comunidad	98
B.3.3 Unirse a una comunidad.....	98
B.3.4 Eliminar una comunidad.....	99
B.3.4 Abandonar una comunidad	100
B.3.5 Crear un contador	100
B.3.6 Eliminar un contador	101
B.3.7 Modificar valores de un contador	101
B.3.8 Acciones sobre un contador	102
B.3.9 Ajustes de una comunidad	102
B.3.9 Los filtros	102
B.3.9 Acciones sobre la lista de contadores	104
B.3.10 Generación de los informes	105
B.3.11 Estadísticas	107
B.3.12 Valores de la comunidad	107

Índice de tablas

Tabla 1 Requisitos funcionales	13
Tabla 2 Requisitos no funcionales	14
Tabla 3 CU Inicio de sesión con Google	16
Tabla 4 CU Cerrar sesión.....	17
Tabla 5 CU Cambiar cuenta de usuario	18
Tabla 6 CU Ver la descripción de un parámetro.....	19
Tabla 7 CU Iniciar sesión como usuario	20
Tabla 8 CU Añadir un contador	21
Tabla 9 CU Lista de contadores de usuario	22
Tabla 10 CU Visualizar un contador de usuario.....	23
Tabla 11 CU Visualizar la lista de contadores de las comunidades añadidas	24
Tabla 12 CU Visualizar los datos de un contador	25
Tabla 13 CU Ver la localización de un contador	26
Tabla 14 CU Filtrar la lista de contadores.....	27
Tabla 15 CU Crear un contador	28
Tabla 16 CU Editar un "parámetro de usuario" de un contador	29
Tabla 17 CU Editar un "parámetro de administrador" de un contador	30
Tabla 18 CU Hacer enviar un mensaje al contador del usuario	31
Tabla 19 CU Hacer la fecha visible o no para el usuario.....	32
Tabla 20 CU Eliminar un contador	33
Tabla 21 CU Enviar sugerencias al desarrollador	34
Tabla 22 CU Ver lista de comunidades	35
Tabla 23 CU Crear una comunidad	36
Tabla 24 CU Unirse a una comunidad	37
Tabla 25 CU Eliminar una comunidad	38
Tabla 26 CU Abandonar una comunidad.....	39
Tabla 27 CU Editar los parámetros de una comunidad	40
Tabla 28 CU Editar los ajustes de una comunidad	41
Tabla 29 CU Eliminar las fechas de todos los contadores	42
Tabla 30 CU Eliminar los mensajes de todos los contadores	43
Tabla 31 CU Generar el informe de la comunidad	44
Tabla 32 CU Desplazar la última lectura de todos los contadores	46
Tabla 33 CU Visualizar las estadísticas de los contadores.....	47
Tabla 34 Caso de prueba TST1.....	73
Tabla 35 Caso de prueba TST2.....	73
Tabla 36 Caso de prueba TST3.....	73
Tabla 37 Caso de prueba TST4.....	73
Tabla 38 Caso de prueba TST5.....	74
Tabla 39 Caso de prueba TST6.....	74
Tabla 40 Caso de prueba TST7.....	74

Tabla 41 Caso de prueba TST8	74
Tabla 42 Caso de prueba TST9	75
Tabla 43 Caso de prueba TST10	75
Tabla 44 Caso de prueba TST11	75
Tabla 45 Caso de prueba TST12	75
Tabla 46 Caso de prueba TST13	76
Tabla 47 Caso de prueba TST14	76
Tabla 48 Caso de prueba TST15	76
Tabla 49 Caso de prueba TST16	76
Tabla 50 Caso de prueba TST17	77
Tabla 51 Caso de prueba TST18	77
Tabla 52 Caso de prueba TST19	77
Tabla 53 Caso de prueba TST20	77
Tabla 54 Caso de prueba TST21	78
Tabla 55 Caso de prueba TST22	78
Tabla 56 Caso de prueba TST23	78
Tabla 57 Caso de prueba TST24	78
Tabla 58 Caso de prueba TST25	79
Tabla 59 Caso de prueba TST26	79
Tabla 60 Caso de prueba TST27	79
Tabla 61 Caso de prueba TST28	79
Tabla 62 Caso de prueba TST29	80
Tabla 63 Caso de prueba TST30	80
Tabla 64 Caso de prueba TST31	80
Tabla 65 Caso de prueba TST32	80
Tabla 66 Caso de prueba TST33	81

Índice de figuras

Figura 1 Alberca de regadío para al abastecimiento de los árboles	1
Figura 2 Versiones de Android	1
Figura 3 Versiones de IOS	2
Figura 4 Logo Java.....	5
Figura 5 Logo Magicdraw	5
Figura 6 Logo Firebase	5
Figura 7 Logo Android Studio	6
Figura 8 Logo Gradle.....	6
Figura 9 Logo Draw.io	6
Figura 10 Logo Trello	6
Figura 11 Logo Git.....	6
Figura 12 Logo Bitbucket	7
Figura 13 Casos de uso	15
Figura 14 Mockup inicio de sesión	16
Figura 15 Mockup lista usuario/administrador	17
Figura 16 Mockup inicio de sesión	18
Figura 17 Mockup contador usuario	19
Figura 18 Mockup selección del usuario	20
Figura 19 Mockup agregar contador	21
Figura 20 Mockup lista usuario	22
Figura 21 Mockup contador usuario	23
Figura 22 Mockup lista administrador.....	24
Figura 23 Mockup contador administrador	25
Figura 24 Mockup mapa contador	26
Figura 25 Mockup filtro	27
Figura 26 Mockup crear contador	28
Figura 27 Mockup editar parámetro usuario	29
Figura 28 Mockup editar parámetro administrador	30
Figura 29 mockup mensaje privado	31
Figura 30 Mockup fecha visible	32
Figura 31 Mockup eliminar contador	33
Figura 32 Mockup contactar con el desarrollador	34
Figura 33 Mockup lista comunidades.....	35
Figura 34 Mockup crear comunidad.....	36
Figura 35 Mockup unirse a comunidad	37
Figura 36 Mockup eliminar comunidad.....	38
Figura 37 Mockup abandonar comunidad	39
Figura 38 Mockup comunidad	40
Figura 39 Mockup comunidad	41
Figura 40 Mockup comunidad	42

Figura 41 Mockup comunidad.....	43
Figura 42 Mockup informe comunidad.....	44
Figura 43 Mockup comunidad.....	46
Figura 44 Mockup estadísticas	47
Figura 45 Metodología scrum	49
Figura 46 Proceso sprint.....	50
Figura 47 Tablón Trello.....	51
Figura 48 Estructura Cloud Firestore	54
Figura 49 Tipos de datos Cloud Firestore.....	54
Figura 50 Consola de datos de Cloud Firestore	55
Figura 51 Modelo de datos Cloud Firestore.....	55
Figura 52 Diseño de la base de datos.....	59
Figura 53 Diagrama de la base de datos	60
Figura 54 Diseño de la interfaz.....	60
Figura 55 Diseño del mockup	61
Figura 56 Uso del mockup	61
Figura 57 Diagrama de clases.....	63
Figura 58 Diagrama de paquetes	64
Figura 59 Archivo google-services.json	67
Figura 60 Firebase rules	68
Figura 61 Emulador Android Studio	70
Figura 62 Diagrama de Test Lab	72
Figura 63 Pantalla habilitar orígenes desconocidos.....	88
Figura 64 Pantalla de instalación de la aplicación.....	88
Figura 65 Habilitar modo desarrollador	89
Figura 66 Ejecución de la aplicación en Android Studio	89
Figura 67 Rules Cloud Firestore para MyContador	90
Figura 68 Descripción caso de uso MyContador	93
Figura 69 Interacción entre usuario y administrador	94
Figura 70 Inicio de sesión en la app	94
Figura 71 Añadir un contador	95
Figura 72 Datos de un contador	96
Figura 73 Eliminar un contador	97
Figura 74 Mensajes del administrador.....	97
Figura 75 Creación de una comunidad.....	98
Figura 76 Pasos para unirse a una comunidad.....	99
Figura 77 Mensaje de borrar comunidad.....	99
Figura 78 Mensaje de abandonar comunidad	100
Figura 79 Ventana de creación de un contador	100
Figura 80 Mensaje de eliminación de un contador.....	101
Figura 81 Modificación del valor de un contador	101
Figura 82 Ajustes de una comunidad	102
Figura 83 Filtros de la aplicación	103
Figura 84 Acciones de una comunidad	104
Figura 85 mensaje de borrado de fechas.....	104
Figura 86 Mensaje de borrado de mensajes.....	104
Figura 87 Mensaje de nuevo ejercicio	105

Figura 88 Pasos para la generación de informes.....	106
Figura 89 Informe generado en Excel.....	106
Figura 90 Estadísticas de la comunidad.....	107
Figura 91 Valores de una comunidad	108

1

Introducción

1.1 Motivación

La motivación de este trabajo es poder crear un proyecto real, en el que ayudar a un colectivo, en este caso, los encargados de administrar los contadores.

La aplicación está diseñada para que los clientes sepan el estado de su contador, pero un principal objetivo es el de ayudar a los encargados a poder tener un registro de los contadores y sus mediciones para poder generar el informe y tramitarlo a la compañía de forma más fácil y efectiva.

Uno de los colectivos son, los encargados de medir los niveles de agua consumida de las albercas de los campesinos que estos usan para el regadío.



Figura 1 Alberca de regadío para el abastecimiento de los árboles

1.2 Objetivos

El objetivo consiste en el desarrollo de una aplicación Android cuya función es facilitar la tarea del revisor a la hora de gestionar de todos los contadores y permitir a los usuarios tener una estimación de su próxima factura antes de recibirla y ver los datos de su contador.

La aplicación tendrá dos tipos de usuarios:

- Usuario, el cual puede agregar un contador a su lista de contadores mediante el número de contador y una contraseña suministrada por el administrador.
- Administrador, el cual posee acceso a todos los contadores, así como los parámetros del contador y datos adicionales de este que un usuario normal no puede ver.

Otras características son:

- Mediante la aplicación el administrador actualizará las mediciones de los contadores para que los datos de estos sean visibles por los usuarios, así como otras variables globales con los que la aplicación hará otros cálculos como el precio del agua/luz/gas.
- El usuario podrá, además de ver su consumo, ver otros datos, como avisos del administrador, estado del contador, lecturas actuales y anteriores, límite de consumo, fuente del suministro (agua, gas, luz), etc.
- La autenticación será mediante Google y se añadirán contadores mediante su número y contraseña.
- El administrador podrá hacer un análisis de todos los contadores que tiene asignados.
- La aplicación avisará a los usuarios mediante notificaciones con mensajes como revisiones, cortes de agua, etc.
- El administrador podrá descargar un informe de los contadores para su empresa.

1.3 Estructura de la memoria

La memoria se ha estructurado en función de las distintas fases que se han seguido para la realización del desarrollo de la aplicación, las cuales son las siguientes:

1. Introducción

Definición y explicación de los distintos episodios contenidos en la memoria correspondientes a las diferentes secciones en las que se ha definido el proyecto.

2. Sistemas Operativos Móviles

Se definen los distintos sistemas operativos móviles y los motivos por lo que se ha optado por desarrollar la aplicación en uno solo de ellos.

3. Tecnologías usadas

Se definen las distintas tecnologías que han sido necesarias durante el desarrollo, operación y mantenimiento del software desarrollado.

4. Librerías usadas

Se especifican las librerías que han sido necesarias para que el proyecto cumpliera los requisitos requeridos, aunque se han usado más librerías además de las mencionadas, estas no se han acabado usando debido a incompatibilidades o por sustitución de una librería mejor, ya que una solución puede abordarse con varias librerías, por lo que se han intentado seleccionar las que mejor cumplen los requisitos.

5. Análisis

En esta sección se especifican la obtención de requisitos a partir del cliente para la realización del proyecto software.

6. Metodología de desarrollo

En esta sección se detalla la metodología de trabajo, así como el entorno de desarrollo y las distintas complicaciones y limitaciones a la hora de obtener los requisitos por parte del cliente.

7. Bases de datos

Se definen las distintas bases de datos que se han tenido en consideración y la motivación por la decisión de la base de datos seleccionada así como su funcionamiento.

8. Diseño de la aplicación

Esta sección contiene la descripción y presentación del mockup así como los distintos diseños realizados sobre la aplicación de la interfaz la estructura usada para satisfacer los requisitos del cliente y usada para el desarrollo.

9. Desarrollo de la aplicación

En esta sección se detallan las distintas fases sobre el desarrollo de la aplicación, así como las características, estructura y funcionamiento del código implementado para el funcionamiento de la aplicación.

10. Seguridad

Esta sección contiene la descripción del análisis de seguridad de la aplicación, sus posibles puntos débiles y sus puntos fuertes.

11. Pruebas

En esta sección se especifican las distintas pruebas por los que ha pasado el software desarrollado para su correcto funcionamiento, así como los distintos métodos desarrollados para la ejecución de dichas pruebas.

12. Conclusiones

En esta sección se detallan las conclusiones finales a las que se ha llegado tras la finalización del proyecto, así como sus posibles líneas futuras.

Apéndice A Manual de instalación

En este apéndice se detallan las diferentes etapas necesarias para la realización de la instalación del software.

Apéndice B: Manual de usuario

En este apéndice se detallan todas las funcionalidades de la aplicación que pueden realizar los usuarios, así como los pasos que hay que realizar para poder llevar a cabo cada una de ellas.

2

Sistemas Operativos Móviles

2.1 Android

Android es un sistema operativo móvil basado en núcleo Linux y otros software de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas, relojes inteligentes (Wear OS), automóviles (Android Auto) y televisores (Android TV). (Wikipedia, s.f.)

2.1.1 Versiones

Desde su primera versión en 2001, Android ha ido sacando nuevas versiones donde ha ido incluyendo distintas mejoras, estas versiones están asociadas a un nivel de API las cuales son las que usamos cuando implementamos una aplicación en Android para ver si será compatible o no con el código implementado.

Nombre código ↕	Número de versión ↕	Fecha de lanzamiento ↕	Nivel de API ↕
Apple Pie ⁵⁵	1.0	23 de septiembre de 2008	1
Banana Bread ⁵⁵	1.1	9 de febrero de 2009	2
Cupcake	1.5	25 de abril de 2009	3
Donut	1.6	15 de septiembre de 2009	4
Eclair	2.0 – 2.1	26 de octubre de 2009	5 – 7
Froyo	2.2 – 2.2.3	20 de mayo de 2010	8
Gingerbread	2.3 – 2.3.7	6 de diciembre de 2010	9 – 10
Honeycomb ⁵⁶	3.0 – 3.2.6	22 de febrero de 2011	11 – 13
Ice Cream Sandwich	4.0 – 4.0.5	18 de octubre de 2011	14 – 15
Jelly Bean	4.1 – 4.3.1	9 de julio de 2012	16 – 18
KitKat	4.4 – 4.4.4	31 de octubre de 2013	19 – 20
Lollipop	5.0 – 5.1.1	12 de noviembre de 2014	21 – 22
Marshmallow	6.0 – 6.0.1	5 de octubre de 2015	23
Nougat	7.0 – 7.1.2	15 de junio de 2016	24 – 25
Oreo	8.0 – 8.1	21 de agosto de 2017	26 – 27
Pie	9.0	6 de agosto de 2018	28

Figura 2 Versiones de Android

2.1.2 Estructura

La estructura de Android se divide en capas, las cuales son:

- **Aplicación:** Es la capa superior donde se encuentra las distintas aplicaciones nativas e instaladas en Android.
- **Marco de trabajo de aplicaciones:** Es la capa donde se encuentran los principales componentes de Android que pueden utilizar los desarrolladores en sus aplicaciones.
- **Bibliotecas:** Son distintas librerías implementadas en C/C++ usadas por android y por los desarrolladores para poder tener acceso a las distintas funcionalidades principales que son necesarias como bibliotecas de gráficos, SQLite, etc.
- **Runtime de Android:** Implementan la mayor parte de las distintas librerías que ofrece java.
- **Núcleo Linux:** Es la capa más conectada al hardware, la cual transforma las distintas llamadas en instrucciones para el procesador y los distintos componentes hardware del teléfono.

2.2 IOS

iOS es un sistema operativo móvil de la multinacional Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), después se ha usado en dispositivos como el iPod touch y el iPad. Apple no permite la instalación de iOS en hardware de terceros. (Wikipedia, s.f.)

2.1.1 Versiones

Su primera versión fue lanzada en 2010 y todas sus actualizaciones se realizan mediante OTA (Over The Air). Las distintas actualizaciones se realizan solo sobre ciertos dispositivos, quedando desactualizados permanentemente los dispositivos mas antiguos.

Build	Versión	Fecha de lanzamiento	Dispositivos con Fin de Ciclo de Vida
7E18	3.1.3	2 de febrero de 2010	iPhone e iPod touch (1.ª generación)
8C148	4.2.1	21 de noviembre de 2010	iPhone 3G e iPod touch (2.ª generación)
9B206	5.1.1	7 de mayo de 2012	iPad (1.ª generación) e iPod touch (3.ª generación)
10B500	6.1.6	23 de febrero de 2014	iPod touch (4.ª generación) e iPhone 3GS
11D257	7.1.2	29 de junio de 2014	iPhone 4
13G36	9.3.6	22 de julio de 2019	iPhone 4S, iPad 2, iPad 3, iPad Mini (1.ª generación) e iPod touch (5.ª generación)
14G60	10.3.4	22 de julio de 2019	iPhone 5, iPhone 5C e iPad (4.ª generación)
16G102	12.4.8	15 de julio de 2020	iPhone 5s, iPhone 6, iPad Air (1.ª generación), iPad Mini (2.ª y 3.ª generación) e iPod touch (6.ª generación)
17G68	13.6	15 de julio de 2020	
18A5319I	14.0.1	7 de julio de 2020	iPhone 6S, iPhone SE (Primera Generación), iPad (5.ª generación), iPad Air 2, iPad Mini (4.ª Generación) y iPad Pro (1.ª Generación)

Figura 3 Versiones de IOS

2.1.2 Estructura

La estructura de IOS se orientan en las siguientes capas:

- **Cocoa Touch:** Es la capa más superior, es una API para la creación de las distintas aplicaciones para IOS.
- **Medios:** Es la capa que proporciona acceso a los distintos ficheros multimedia como videos, gráficos, etc.
- **Core services:** Son los servicios principales que se ofrecen para que los desarrolladores los utilicen en sus aplicaciones como SQLite, acceso a red, etc.
- **Core OS:** Es el núcleo del sistema, el cual es el encargado de conectar la parte del software con el hardware del dispositivo. Está basado en OS X de Apple el cual a su vez está basado en el kernel de Linux.

Tecnologías utilizadas

3.1 Java



Figura 4 Logo Java

Es un lenguaje de programación orientado a objetos el cual fue desarrollado en 1995. Junto con Kotlin, son los dos lenguajes de programación usados para el desarrollo nativo de aplicaciones Android.

3.2 MagicDraw



Figura 5 Logo
Magicdraw

MagicDraw es una herramienta de modelado visual UML , SysML , BPMN y UPDM con soporte de colaboración en equipo. Diseñada para analistas de negocios, analistas de software, programadores e ingenieros de control de calidad, esta herramienta de desarrollo dinámica y versátil facilita el análisis y el diseño de sistemas y bases de datos orientados a objetos (OO).

3.3 Firebase



Figura 6 Logo
Firebase

Es una plataforma web que facilita el desarrollo tanto de aplicaciones web como nativas para Android e iOS. Proporciona diversas herramientas como bases de datos, servicios de almacenamiento de archivos, herramientas para el desarrollo de test, monitorización de las aplicaciones y otras muchas funciones.

3.4 Android Studio



Figura 7 Logo
Android Studio

Es el entorno de desarrollo principal para el desarrollo de aplicaciones nativas en Android. Esta desarrollado por Google y su primera versión fue lanzada en 2014. Soporta el desarrollo de aplicaciones en Java, Kotlin y C++, aunque se pueden usar otros lenguajes como Go con el uso de complementos.

3.5 Gradle



Figura 8 Logo
Gradle

Es la herramienta que utiliza Android Studio la cual es la encargada de compilar el código de la aplicación y construir la app. También es la encargada de la gestión de las dependencias.

3.6 Draw.io



Figura 9 Logo
Draw.io

Es una herramienta de código abierto a partir de la cual podemos hacer distintos tipos de diagramas y mockups.

3.7 Trello



Figura 10 Logo Trello

Es una plataforma utilizada para la gestión de proyectos. La cual tiene una estructura e forma de tableros virtuales a los cuales se le van añadiendo las distintas tareas para mejorar la organización durante la realización de las mismas.

3.8 Git



Figura 11 Logo Git

Es una herramienta pensada para el control de versiones de una aplicación, así como gestionar la integración de las distintas modificaciones ejercidas sobre el código y tener un historial.

3.9 Bitbucket



Es un sistema de control de versiones en la nube el cual está pensado para los proyectos que estén integrados con Mercurial o Git.

Figura 12 Logo
Bitbucket

4

Librerías usadas

4.1 Material Design Icons

Muchos de los iconos de la aplicación han sido obtenidos a partir de la librería oficial de material design, la cual está integrada en Android Studio. material.io/resources/icons

4.2 Apache POI

Es una librería desarrollada por Apache Software Foundation la cual permite crear, editar y visualizar distintos tipos de documentos en los formatos de Microsoft office como Word, PowerPoint, Excel, etc.

Es la librería usada en la app para la elaboración de los distintos informes en formato Excel.

poi.apache.org

4.3 MPAndroidChart

Es una librería de código abierto destinada a la visualización de distintos tipos de gráficos, estos se han utilizado para poder visualizar distintos los datos de las estadísticas de forma dinámica de la app.

github.com/PhilJay/MPAndroidChart

4.4 Firebase

Es la plataforma principal que se ha usado para obtener los distintos **servicios externos** implementados en la aplicación que necesitan un servidor externo.

4.4.1 Firebase Authentication

Es un servicio con el cual se permite la autenticación de los usuarios con distintas cuentas como Google, Facebook, Twitter, etc. Para esta aplicación, se ha utilizado la autenticación con la cuenta de Google. firebase.google.com/docs/auth

4.4.2 Firebase Cloud Firestore

Es una base de datos NoSQL escalable y flexible, la cual permite tener los datos sincronizados en la nube. firebase.google.com/docs/firestore

4.4.3 Firebase Cloud Storage

Es un servicio para el almacenamiento de todo tipo de archivos.

firebase.google.com/docs/storage

4.4.4 Firebase Crashlytics

Es una herramienta que funciona en tiempo real, a partir de la cual, podemos ver los distintos problemas que ha podido surgir en la aplicación durante el tiempo de ejecución. Provee una serie de datos asociados a cada falla encontrada a partir de los cuales se permite encontrar el problema que ha podido originar la excepción para así solucionarla. firebase.google.com/docs/crashlytics

4.4.4 Firebase Analytics

Es el servicio principal de Firebase, el cual integra monitorización anónima de los usuarios para obtener distintos tipos de datos, como por ejemplo, las funciones más usadas de la aplicación, el tiempo medio de uso de esta, etc.

firebase.google.com/docs/analytics

5

Análisis

5.1 Descripción

El proceso de análisis es la etapa más importante del desarrollo de la aplicación, ya que es en esta donde se acuerdan con el cliente todas las características que debe tener la aplicación además, también hay que obtener la documentación sobre el sector para el se desarrolla la aplicación.

5.2 Obtención de requisitos

Para la obtención de requisitos se ha realizado una reunión principal con el cliente, en la cual se ha elaborado un informe general de requisitos principales, a partir de este, mediante un desarrollo ágil, se han ido realizando distintas iteraciones para la corrección de requisitos errores, requisitos adicionales, etc. Se han recogido dos tipos de requisitos:

- Requisitos **funcionales** (Palomo, s.f.): Los requisitos funcionales expresan las tareas que el sistema tiene que hacer para producir resultados útiles para los usuarios. Suelen definirse mediante unos datos de entrada sobre los que se realizan determinados cálculos y generan nuevos datos como salida de dichos cálculos.

- Requisitos **no funcionales** (Palomo, s.f.): Los requisitos no funcionales expresan condiciones o restricciones bajo las que se deben ejecutar o desarrollar el sistema,

definen propiedades o atributos de calidad que se tienen que cumplir cuando el sistema lleva a cabo algunas de las acciones que le piden los usuarios.

5.3 Requisitos funcionales

Los requisitos funcionales recogidos son los siguientes:

ID	Nombre	Descripción
RF1	Iniciar sesión	El usuario podrá iniciar sesión con su cuenta de Google
RF2	Cerrar sesión	El usuario podrá cerrar sesión de su cuenta de Google
RF3	Cambiar de usuario	El usuario podrá cambiar su cuenta de Google
RF4	Iniciar sesión como usuario	El usuario podrá iniciar sesión como usuario
RF5	Iniciar sesión como administrador	El usuario podrá iniciar sesión como administrador
RF6	Añadir un contador	El usuario podrá añadir un contador para que el sistema se lo muestre en su lista
RF7	Visualizar una lista de contadores	El usuario podrá visualizar su lista de contadores
RF8	Visualizar un contador	El usuario podrá visualizar los datos de su contador
RF9	Ver la descripción de los parámetros	El usuario y el administrador verán una breve descripción de los parámetros al seleccionarlos
RF10	Visualizar los contadores de las comunidades	El administrador podrá visualizar los contadores de las comunidades a las que pertenece
RF11	Visualizar un contador	El administrador podrá visualizar los datos de su contador
RF12	Ver la localización de un contador	El administrador podrá ver en un mapa la localización del contador
RF13	Filtrar los contadores	El administrador podrá filtrar la lista de contadores por diferentes filtros simultáneamente.
RF14	Crear un contador	El administrador podrá crear un contador para una comunidad
RF15	Editar los parámetros de un contador	El administrador podrá editar los parámetros que no son generados automáticamente de un contador
RF16	Enviar un mensaje al dueño de un contador	El usuario podrá enviar un mensaje privado a todos los dueños de un contador

RF17	Hacer la fecha de un contador visible	El administrador podrá decidir que la fecha sea o no visible en el contador
RF18	Eliminar un contador	El administrador podrá eliminar un contador
RF19	Enviar sugerencias al desarrollador	El administrador podrá enviar sugerencias al desarrollador mediante un formulario
RF20	Ver la lista de comunidades	El administrador podrá ver la lista de todas las comunidades en las que este
RF21	Crear una comunidad	El administrador podrá crear una comunidad
RF22	Unirse a una comunidad	El administrador podrá unirse a una comunidad
RF23	Eliminar una comunidad	El administrador podrá eliminar una comunidad si el la creo
RF24	Abandonar una comunidad	El administrador podrá abandonar una comunidad de la cual no sea el dueño
RF25	Editar los parámetros generales de una comunidad	El administrador podrá editar los parámetros generales de una comunidad
RF26	Editar los ajustes de una comunidad	El administrador podrá cambiar los ajustes generales de una comunidad
RF27	Eliminar las fechas de todos los contadores	El administrador podrá eliminar las fechas de todos los contadores de una comunidad
RF28	Eliminar los mensajes de todos los contadores	El administrador podrá eliminar los mensajes de todos los contadores de una comunidad
RF29	Generar un informe de todos los contadores	El administrador podrá generar un informe de todos los contadores de una comunidad
RF30	Desplazar la fecha de lectura de todos los contadores	El administrador podrá desplazar la fecha de lectura de todos los contadores
RF31	Visualizar las estadísticas de los contadores	El administrador podrá visualizar las estadísticas de los contadores de una comunidad

Tabla 1 Requisitos funcionales

5.4 Requisitos no funcionales

Los requisitos funcionales recogidos son los siguientes:

ID	Nombre	Descripción
RNF1	Soporte para múltiples idiomas	La aplicación deberá tener soporte para múltiples idiomas, incluyendo como mínimo inglés y español
RNF2	Seguridad en los datos	Los datos de los usuarios deben de ser accedidos de forma segura
RNF3	Uso de la app sin conexión	La aplicación debe de poder usarse incluso si el dispositivo no tiene conexión con el servidor
RNF4	Uso óptimo de la BBDD	La app debe enviar el mínimo número de consultas a la base de datos
RNF5	Solución efectiva de errores	Cuando la app se cierra debido a un error, este y los motivos se enviarán al desarrollador para su revisión.
RNF6	Actualizaciones automáticas	La app detectará cuando exista una nueva actualización y se actualizará automáticamente.
RNF7	Acceso concurrente	El acceso de datos importantes a la base de datos será forma concurrente
RNF8	Respuesta del servicio	La app tardará como mucho 3 segundos en obtener los datos que se le soliciten
RNF9	Elementos con descripción	Cada acción o elemento deberá tener una descripción sobre el mismo

Tabla 2 Requisitos no funcionales

5.5 Casos de uso

Un **caso de uso** (Wikipedia, s.f.) es la descripción de una acción o actividad. Un diagrama de caso de uso es una descripción de las actividades que deberá realizar alguien o algo para llevar a cabo algún proceso. Los personajes o entidades que participarán en un diagrama de caso de uso se denominan actores.

5.5.1 Diagramas de casos de uso

Los casos de uso se han creado mediante la herramienta draw.io, los casos de uso contemplados son los siguientes:

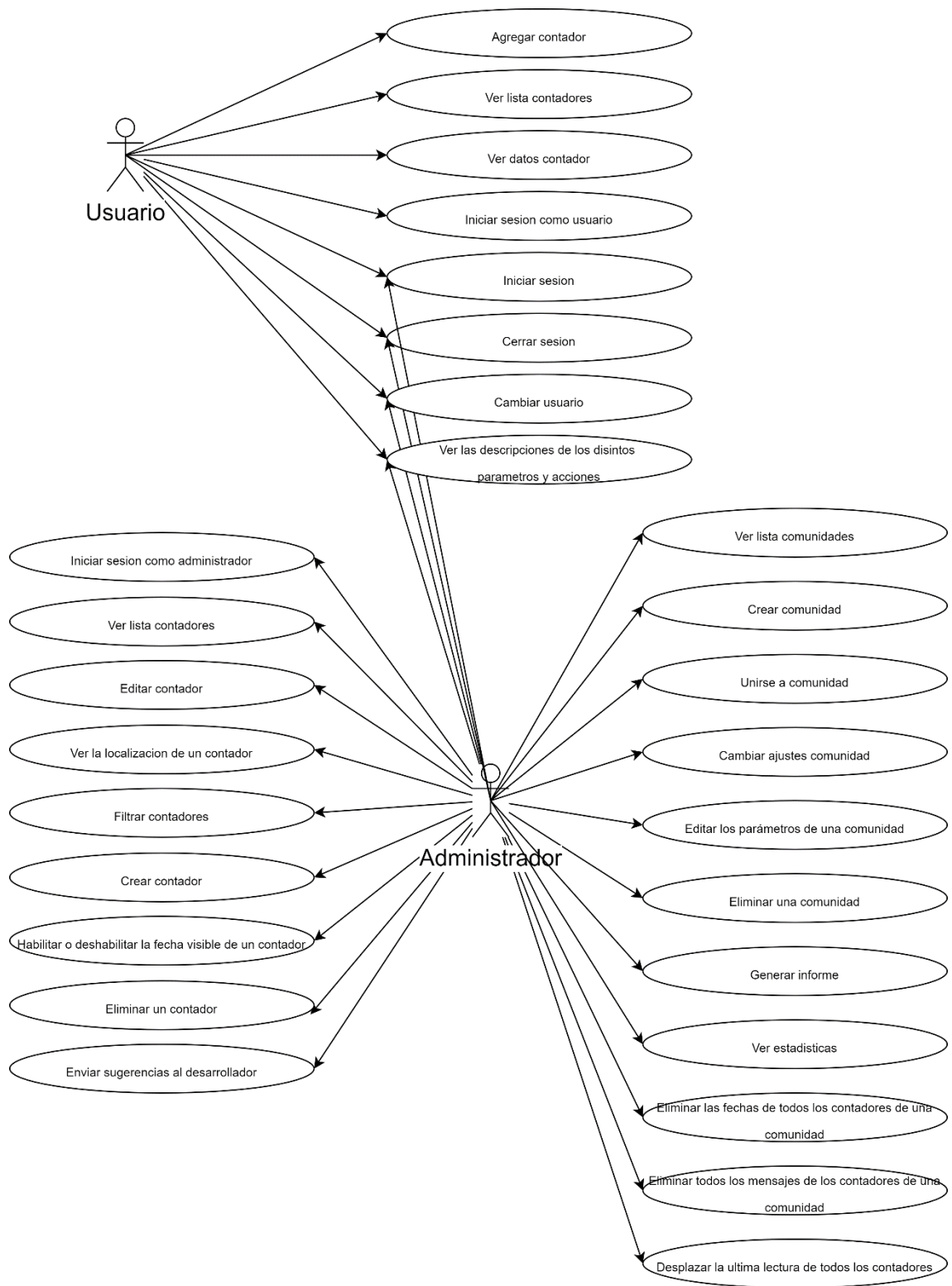


Figura 13 Casos de uso

5.5.2 Lista de casos de uso

Se han especificado los siguientes casos de uso:

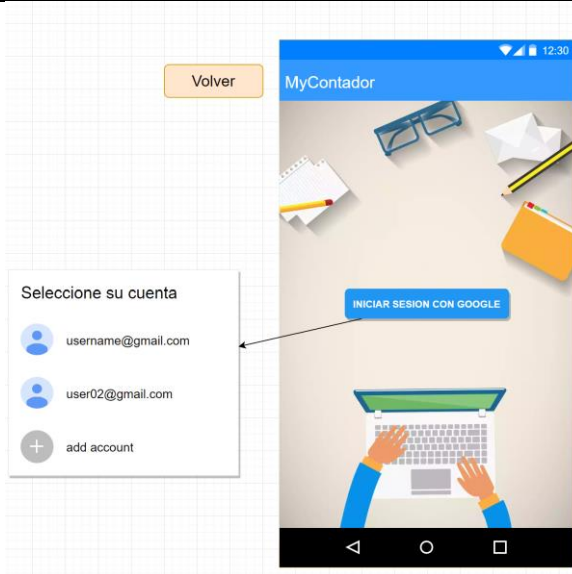
Nombre	Inicio de sesión con Google
Descripción	Un usuario o un administrador deberá poder iniciar sesión en la app con su cuenta de Google
Autor	Usuario o Administrador
Pre-Condición	No hay un usuario logueado
Post-Condición	El usuario esta logueado
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. Selecciona "iniciar sesión con google" 3. Selecciona su cuenta de google 4. Accede a la aplicación con su usuario de google 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. Selecciona "iniciar sesión con google" 3. No selecciona ninguna cuenta 4. Vuelve a la ventana de inicio de sesión 	
Maquetas de interfaz	
 <p>Figura 14 Mockup inicio de sesión</p>	

Tabla 3 CU Inicio de sesión con Google

Nombre	Cerrar sesión
Descripción	Un usuario o un administrador deberá poder cerrar sesión en la app de su cuenta de Google

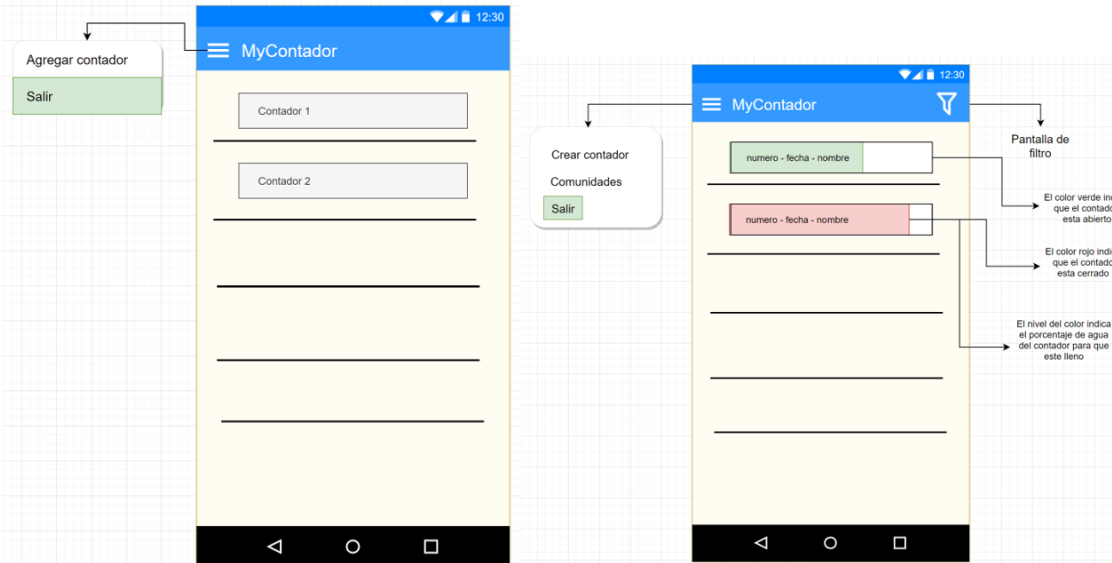
Autor	Usuario o Administrador
Pre-Condición	El usuario esta logueado
Post-Condición	No hay un usuario logueado
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. Selecciona el icono de opciones 3. Selecciona "cerrar sesión" 4. La aplicación muestra la ventana de inicio de sesión 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. Selecciona el icono de opciones 3. Selecciona de nuevo el icono de opciones 4. La aplicación muestra la ventana en la que estaba 	
Maquetas de interfaz	
 <p>Figura 15 Mockup lista usuario/administrador</p>	

Tabla 4 CU Cerrar sesión

Nombre	Cambiar cuenta de usuario
Descripción	Un usuario o un administrador deberá poder cambiar su cuenta de Google
Autor	Usuario o Administrador
Pre-Condición	El usuario esta logueado con una cuenta de Google
Post-Condición	El usuario esta logueado con una cuenta de Google distinta
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. Selecciona el icono de opciones 	

3. Selecciona "cerrar sesión"
4. La aplicación muestra la ventana de inicio de sesión
5. El usuario selecciona "iniciar sesión"
6. El usuario selecciona la otra cuenta de Google
7. La aplicación inicia sesión con la otra cuenta de Google

Flujo alternativo

1. El usuario abre la app
2. Selecciona el icono de opciones
3. Selecciona "cerrar sesión"
4. La aplicación muestra la ventana de inicio de sesión
5. El usuario selecciona "iniciar sesión"
6. El usuario selecciona la misma cuenta de Google
7. La aplicación inicia sesión con la misma cuenta de Google

Maquetas de interfaz

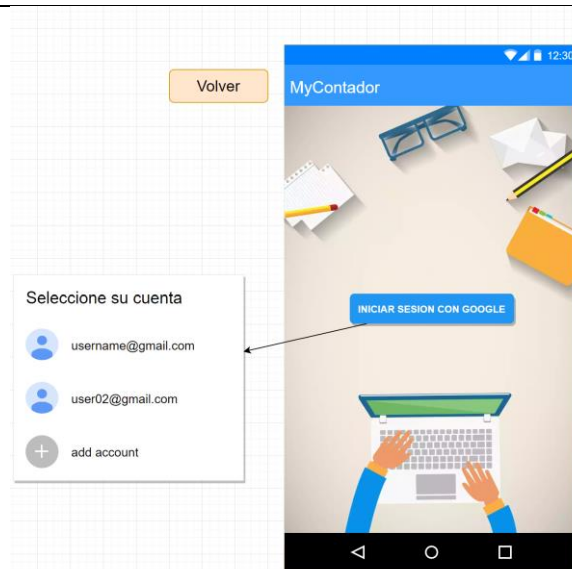


Figura 16 Mockup inicio de sesión

Tabla 5 CU Cambiar cuenta de usuario

Nombre	Ver la descripción de un parámetro
Descripción	Un usuario o un administrador deberá poder ver la descripción de cualquier parámetro o acción
Autor	Usuario o Administrador
Pre-Condición	El usuario o administrador debe estar logueado
Post-Condición	Se muestra una descripción
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario o administrador abre la app 2. Selecciona un contador 3. Selecciona un parámetro 4. La aplicación muestra la ventana con la descripción 	

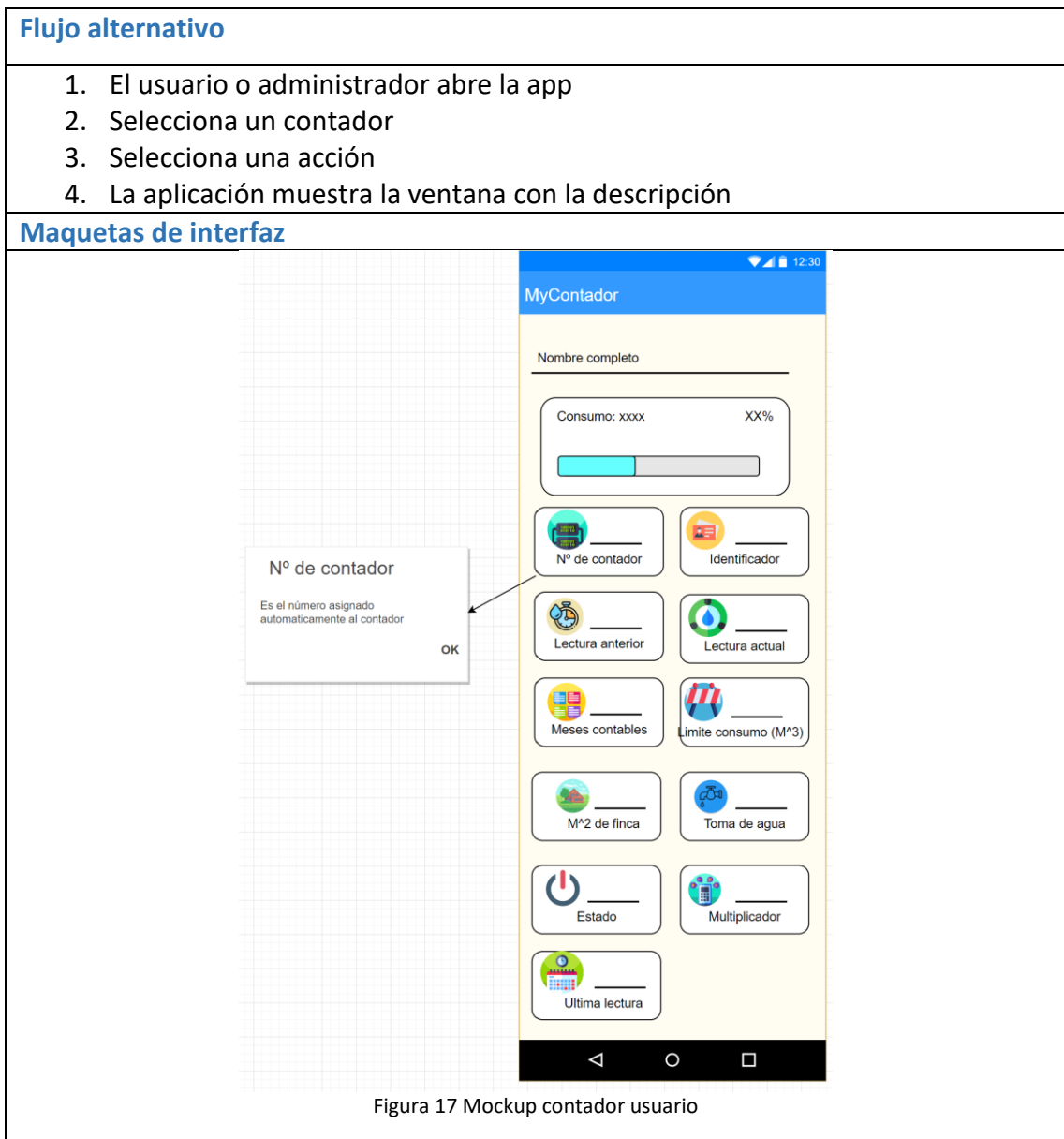


Tabla 6 CU Ver la descripción de un parámetro

Nombre	Iniciar sesión como usuario
Descripción	Un usuario podrá iniciar sesión con el perfil de usuario
Autor	Usuario
Pre-Condición	El usuario logueado sin haber seleccionado su perfil
Post-Condición	Se muestra la interfaz de usuario
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. Selecciona el perfil de usuario 3. La aplicación muestra la interfaz de usuario 	
Flujo alternativo	

1. El usuario abre la app
2. El usuario cierra la app
3. El usuario abre la app
4. La aplicación muestra la interfaz de seleccionar usuario

Maquetas de interfaz



Figura 18 Mockup selección del usuario

Tabla 7 CU Iniciar sesión como usuario

Nombre	Añadir un contador
Descripción	Un usuario podrá añadir un contador si cuenta con su id y contraseña
Autor	Usuario
Pre-Condición	El usuario logueado como usuario
Post-Condición	El contador añadido en la lista
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario selecciona el menú 2. El usuario pulsa la opción "añadir contador" 3. El usuario Introduce el id del contador 4. El usuario introduce la contraseña de contador 5. El usuario pulsa "añadir" 6. El contador se añade a la lista 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. El usuario selecciona el menú 2. El usuario pulsa la opción "añadir contador" 3. El usuario Introduce el id del contador 4. El usuario introduce una contraseña incorrecta 5. El usuario pulsa "añadir" 	

6. El sistema indica que el contador no existe

Maquetas de interfaz

Figura 19 Mockup agregar contador

Tabla 8 CU Añadir un contador

Nombre	Lista de contadores de usuario
Descripción	Un usuario podrá visualizar una lista con todos sus contadores
Autor	Usuario
Pre-Condición	El usuario logueado como usuario
Post-Condición	El sistema muestra la lista de contadores del usuario
Flujo normal	
1. El usuario abre la aplicación 2. El usuario selecciona un contador 3. La aplicación muestra los datos del contador	
Flujo alternativo	

Maquetas de interfaz

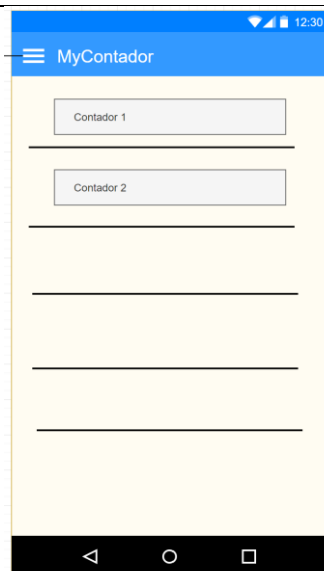


Figura 20 Mockup lista usuario

Tabla 9 CU Lista de contadores de usuario

Nombre	Visualizar un contador de usuario
Descripción	Un usuario podrá visualizar los datos de un contador que seleccione
Autor	Usuario
Pre-Condición	El usuario logueado como usuario
Post-Condición	El sistema muestra los datos de un contador
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la aplicación 2. Se muestra una lista con los contadores añadidos por el usuario 	
Flujo alternativo	

Maquetas de interfaz



Figura 21 Mockup contador usuario

Tabla 10 CU Visualizar un contador de usuario

Nombre	Visualizar la lista de contadores de las comunidades añadidas
Descripción	El administrador podrá ver una lista con todos los contadores de las comunidades a las que pertenezca
Autor	administrador
Pre-Condición	El usuario logueado como administrador
Post-Condición	El sistema muestra la lista de contadores
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. La aplicación muestra la lista e contadores 	
Flujo alternativo	

Maquetas de interfaz

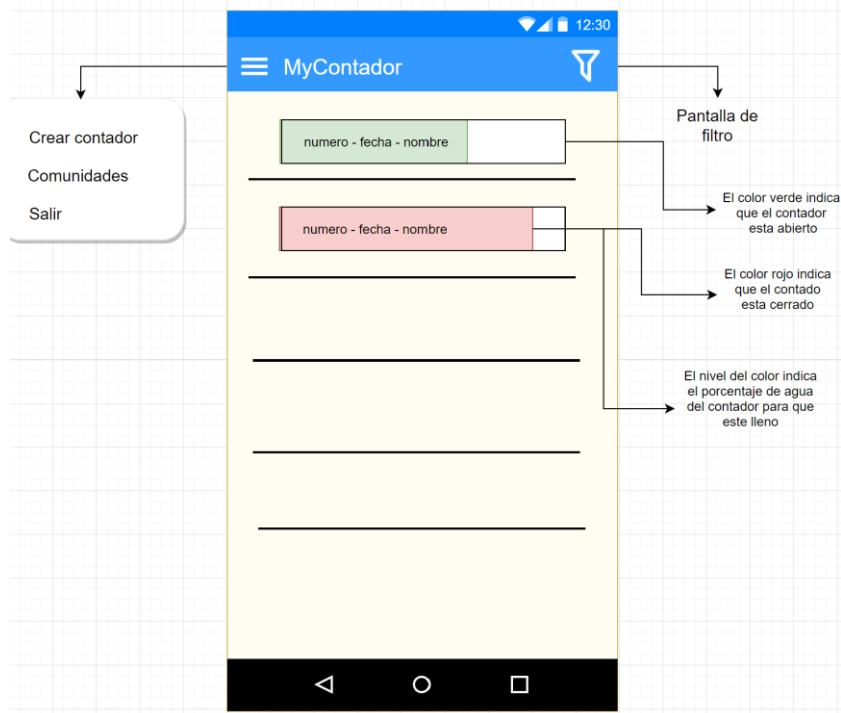


Figura 22 Mockup lista administrador

Tabla 11 CU Visualizar la lista de contadores de las comunidades añadidas

Nombre	Visualizar los datos de un contador
Descripción	El administrador podrá ver todos los datos asociados a un contador
Autor	administrador
Pre-Condición	El usuario logueado como administrador
Post-Condición	El sistema muestra la interfaz del contador
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario selecciona el contador 3. La aplicación muestra la interfaz del contador 	
Flujo alternativo	

Maquetas de interfaz

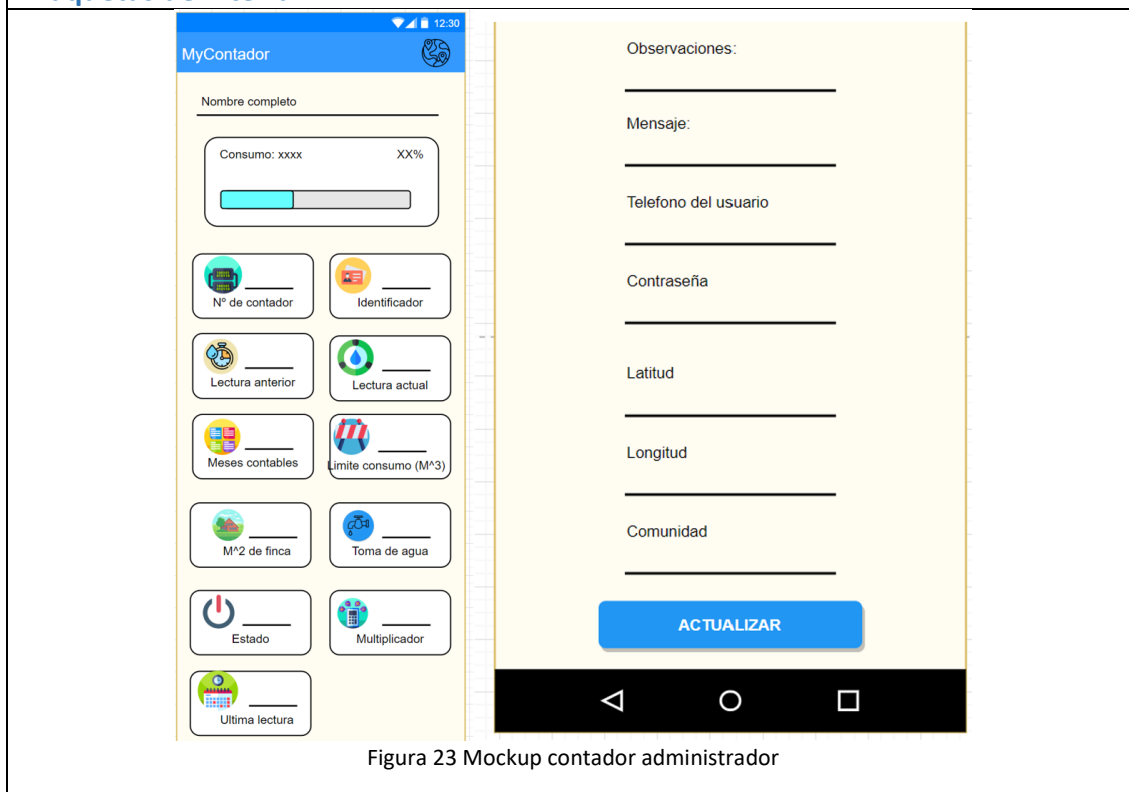


Figura 23 Mockup contador administrador

Tabla 12 CU Visualizar los datos de un contador

Nombre	Ver la localización de un contador
Descripción	El administrador podrá ver la ubicación en un mapa de un contador
Autor	administrador
Pre-Condición	El usuario logueado como administrador
Post-Condición	El sistema muestra la ubicación del contador en el mapa
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario selecciona el contador 3. El usuario selecciona el menú de opciones 4. El usuario selecciona "ver en el mapa" 5. La aplicación muestra un mapa con la ubicación del contador 	
Flujo alternativo	

Maquetas de interfaz

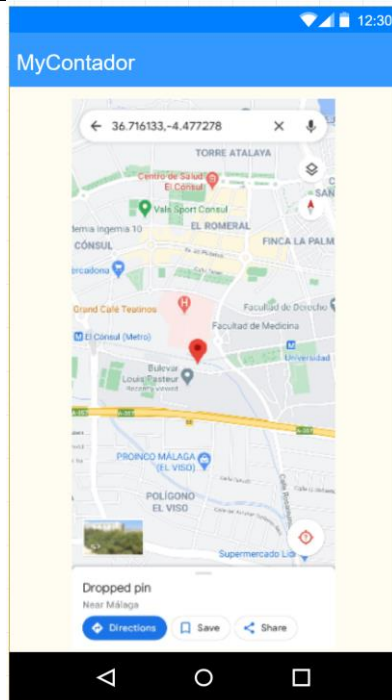


Figura 24 Mockup mapa contador

Tabla 13 CU Ver la localización de un contador

Nombre	Filtrar la lista de contadores
Descripción	El administrador podrá filtrar la lista de contadores que visualiza
Autor	administrador
Pre-Condición	El usuario logueado como administrador
Post-Condición	El sistema muestra la lista de contadores filtrada
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario selecciona el botón de filtrar 3. El usuario selecciona activa el filtro que desea 4. El usuario introduce los valores necesarios en el filtro 5. El usuario pulsa "filtrar" 6. La aplicación muestra la lista de contadores con el filtro 7. La aplicación resalta el botón de "filtrar" 	
Flujo alternativo	
<p>Flujo Alternativo B</p> <ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario selecciona el botón de filtrar 3. El usuario introduce los valores necesarios en el filtro 4. El usuario pulsa "filtrar" 5. La aplicación muestra la lista de todos los contadores 	

Flujo alternativo C

1. El usuario abre la app
2. El usuario selecciona el botón de filtrar
3. El usuario selecciona activa el filtro que desea
4. El usuario pulsa "filtrar"
5. La aplicación muestra la lista de todos los contadores
6. La aplicación resalta el botón de "filtrar"

Flujo alternativo D

1. El usuario abre la app
2. El usuario selecciona el botón de filtrar
3. El usuario selecciona activa el filtro que desea
4. El usuario introduce los valores necesarios en el filtro
5. El usuario vuelve a la ventana anterior
6. La aplicación muestra la lista de contadores con el filtro
7. La aplicación resalta el botón de "filtrar"

Maquetas de interfaz

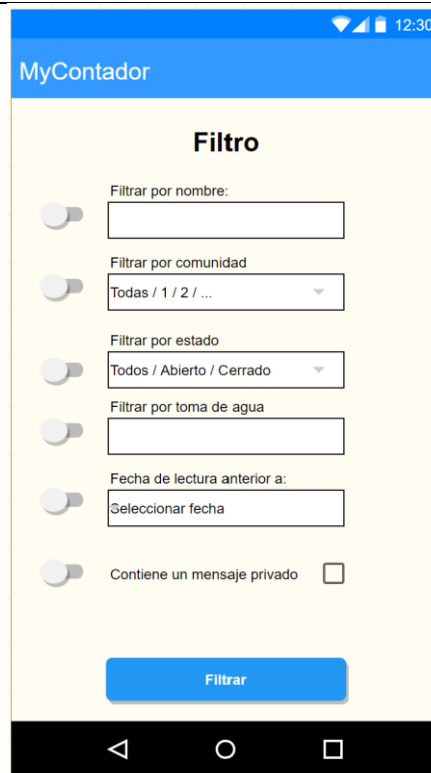


Figura 25 Mockup filtro

Tabla 14 CU Filtrar la lista de contadores

Nombre	Crear un contador
Descripción	El administrador podrá crear un contador asociado a una comunidad
Autor	administrador
Pre-Condición	El usuario logueado como administrador

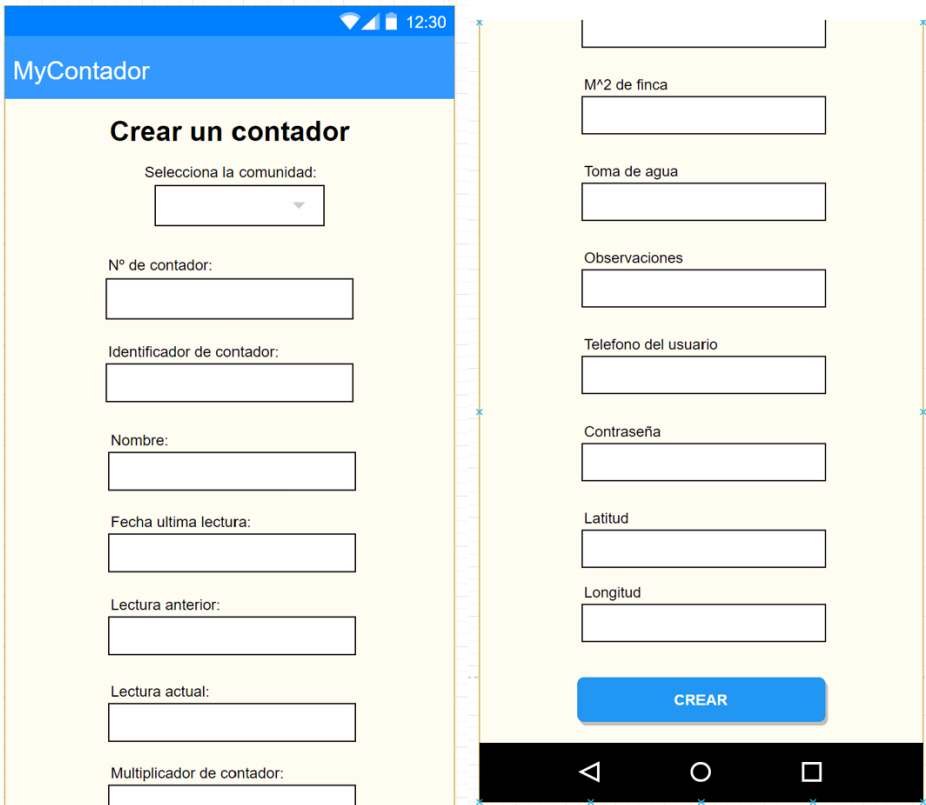
Post-Condición	El contador se añade a la lista
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario selecciona el menú de opciones 3. El usuario selecciona "Crear contador" 4. El usuario selecciona la comunidad 5. El usuario introduce el identificador 6. El usuario pulsa sobre "crear contador" 7. Añade el nuevo contador a la comunidad 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario selecciona el menú de opciones 3. El usuario selecciona "Crear contador" 4. El usuario introduce el identificador 5. El usuario pulsa sobre "crear contador" 6. El sistema muestra el mensaje de error "Seleccione la comunidad" 	
Maquetas de interfaz	
 <p>Figura 26 Mockup crear contador</p>	

Tabla 15 CU Crear un contador

Nombre	Editar un "parámetro de usuario" de un contador
Descripción	El administrador podrá editar un parámetro "editable" del contador

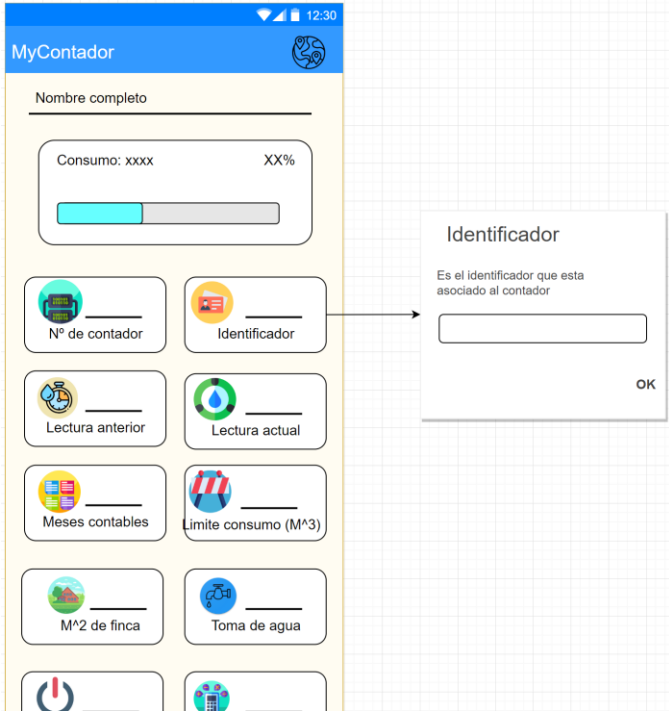
Autor	administrador
Pre-Condición	El usuario logueado como administrador
Post-Condición	El parámetro de contador se modifica
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario selecciona el contador 3. El usuario selecciona el parámetro a editar 4. El sistema muestra un pop-up con la descripción y el valor actual 5. El usuario introduce el nuevo valor 6. El usuario pulsa "Ok" 7. El sistema modifica el parámetro 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario selecciona el contador 3. El usuario selecciona el parámetro a editar 4. El sistema muestra un pop-up con la descripción y el valor actual 5. El usuario introduce el nuevo valor 6. El usuario pulsa "Cancelar" 7. El sistema no modifica el parámetro 	
Maquetas de interfaz	
 <p>The mockup shows the 'MyContador' app interface. At the top, there's a status bar with signal, battery, and time (12:30). Below the app title, there's a section for 'Nombre completo' with a text input field. A progress bar shows 'Consumo: xxxx' and 'XX%'. Below this, there are several parameter cards: 'N° de contador', 'Identificador', 'Lectura anterior', 'Lectura actual', 'Meses contables', 'Limite consumo (M^3)', 'M^2 de finca', and 'Toma de agua'. An arrow points from the 'Identificador' card to a dialog box titled 'Identificador' which contains the text 'Es el identificador que esta asociado al contador', an input field, and an 'OK' button.</p>	

Figura 27 Mockup editar parámetro usuario

Tabla 16 CU Editar un "parámetro de usuario" de un contador

Nombre	Editar un "parámetro de administrador" de un contador
---------------	---

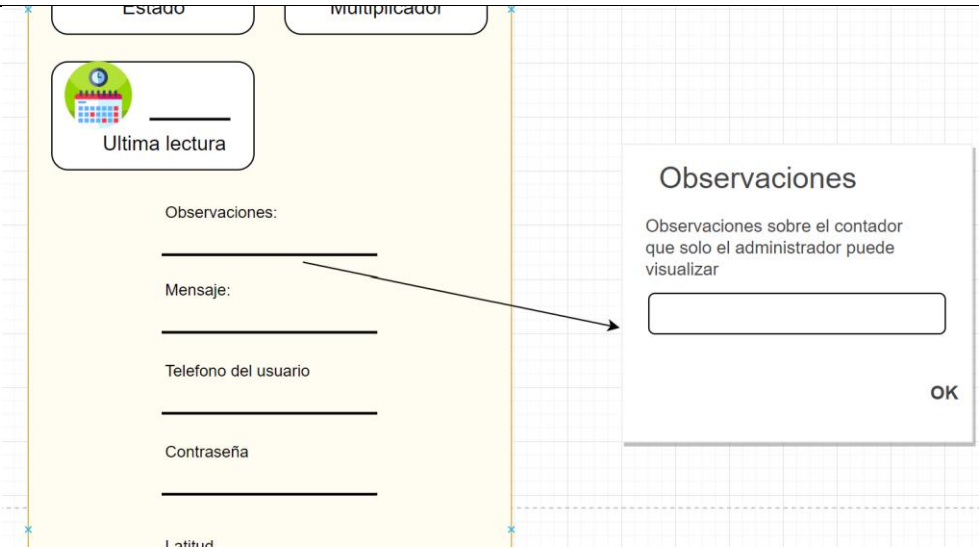
Descripción	El administrador podrá editar un parámetro "del administrador" del contador
Autor	administrador
Pre-Condición	El usuario logueado como administrador
Post-Condición	El parámetro de contador se modifica
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario selecciona el contador 3. El usuario selecciona el parámetro a editar 4. El sistema muestra un pop-up con la descripción y el valor actual 5. El usuario introduce el nuevo valor 6. El usuario pulsa "Ok" 7. El sistema modifica el parámetro 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario selecciona el contador 3. El usuario selecciona el parámetro a editar 4. El sistema muestra un pop-up con la descripción y el valor actual 5. El usuario introduce el nuevo valor 6. El usuario pulsa "Cancelar" 7. El sistema no modifica el parámetro 	
Maquetas de interfaz	
 <p>Figura 28 Mockup editar parámetro administrador</p>	

Tabla 17 CU Editar un "parámetro de administrador" de un contador

Nombre	Hacer enviar un mensaje al contador del usuario
Descripción	El administrador podrá enviar un mensaje al contador del usuario para que este se muestre cuando el usuario visualice el contador.

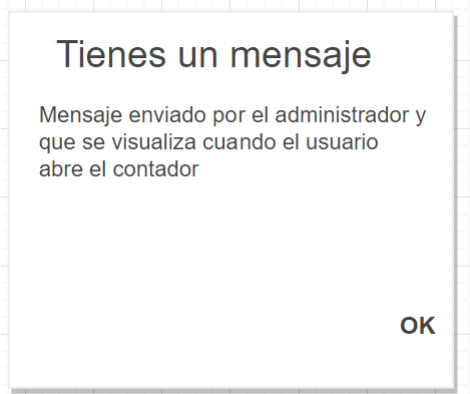
Autor	administrador
Pre-Condición	El usuario logueado como administrador
Post-Condición	Cuando el usuario del contador visualiza el contador se le muestra un mensaje
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario selecciona el contador 3. El usuario selecciona la casilla "mensaje" 4. El usuario introduce el mensaje 5. El usuario pulsa "Ok" 6. El sistema actualiza el valor del campo con el mensaje introducido 7. El sistema le muestra el mensaje al usuario del contador cada vez que este lo visualice 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario selecciona el contador 3. El usuario selecciona la casilla "mensaje" 4. El usuario introduce el mensaje 5. El usuario pulsa "Cancelar" 6. El sistema no modifica el valor del campo "mensaje" 	
Maquetas de interfaz	
 <p>Figura 29 mockup mensaje privado</p>	

Tabla 18 CU Hacer enviar un mensaje al contador del usuario

Nombre	Hacer la fecha visible o no para el usuario
Descripción	El administrador podrá cambiar el comportamiento de un contador para que este le muestre o no la fecha de ultima lectura al usuario
Autor	administrador
Pre-Condición	El usuario logueado como administrador
Post-Condición	La fecha se muestra o no en la vista del contador del usuario

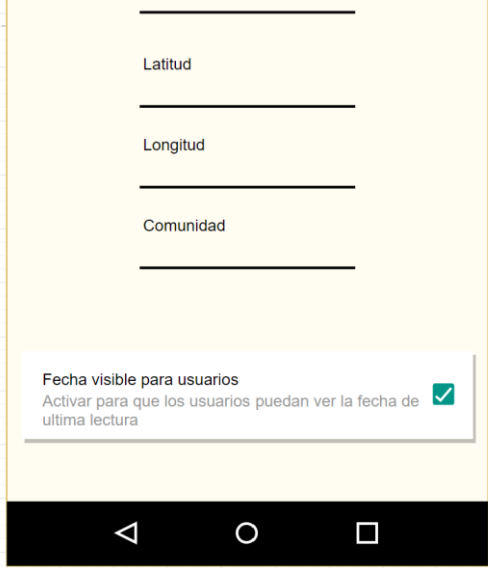
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario selecciona el contador 3. El usuario activa la casilla de "fecha visible" 4. El sistema le muestra la fecha al usuario cuando visualiza el contador 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario selecciona el contador 3. El usuario desactiva la casilla de "fecha visible" 4. El sistema no le muestra la fecha al usuario cuando visualiza el contador 	
Maquetas de interfaz	
 <p>Figura 30 Mockup fecha visible</p>	

Tabla 19 CU Hacer la fecha visible o no para el usuario

Nombre	Eliminar un contador
Descripción	El administrador que además sea el creador de una comunidad podrá eliminar los contadores asociados a esta
Autor	administrador
Pre-Condición	El usuario logueado como administrador El usuario es dueño de la comunidad dele contador
Post-Condición	El contador deja de existir
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario mantiene pulsado el contador 3. El sistema muestra un mensaje indicando si desea eliminarlo 4. El usuario pulsa "Eliminar" 5. El sistema elimina el contador de la comunidad 6. El sistema elimina el contador de la lista 	

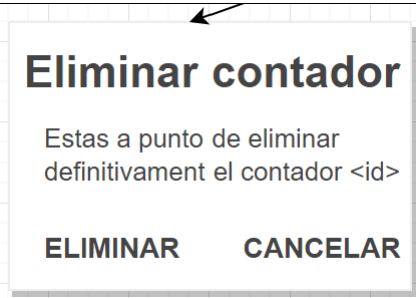

Flujo alternativo
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario mantiene pulsado el contador 3. El sistema muestra un mensaje indicando si desea eliminarlo 4. El usuario pulsa "Cancelar" 5. El sistema no elimina el contador de la comunidad 6. El sistema muestra el contador en la lista
Maquetas de interfaz
 <p>Figura 31 Mockup eliminar contador</p>

Tabla 20 CU Eliminar un contador

Nombre	Enviar sugerencias al desarrollador
Descripción	El administrador podrá mediante un formulario enviar sugerencia sobre la aplicación al desarrollador
Autor	administrador
Pre-Condición	El usuario logueado como administrador
Post-Condición	El desarrollador recibe un mensaje
Flujo normal	
<div><div></div><div><div>1. El usuario abre la app</div><div>2. El usuario pulsa el botón del menú</div><div>3. El usuario selecciona "Contactar"</div><div>4. El sistema muestra una ventana para escribir un mensaje</div><div>5. El usuario introduce el mensaje</div><div>6. El usuario pulsa "enviar"</div><div>7. El sistema envía el mensaje al desarrollador</div><div>8. El sistema muestra una notificación indicando que se ha enviado</div><div>9. El sistema vuelve a mostrar la interfaz anterior</div></div></div>	
Flujo alternativo	
<div><div></div><div><div>1. El usuario abre la app</div><div>2. El usuario pulsa el botón del menú</div><div>3. El usuario selecciona "Contactar"</div><div>4. El sistema muestra una ventana para escribir un mensaje</div><div>5. El usuario introduce el mensaje</div><div>6. El usuario pulsa "cancelar"</div><div>7. El sistema no envía el mensaje al desarrollador</div></div></div>	

8. El sistema vuelve a mostrar la interfaz anterior

Maquetas de interfaz



Contactar con el desarrollador

Indica el mensaje que desea enviar al desarrollador como sugerencias, mejoras, etc.

CANCELAR **ENVIAR**

Figura 32 Mockup contactar con el desarrollador

Tabla 21 CU Enviar sugerencias al desarrollador

Nombre	Ver lista de comunidades
Descripción	El administrador podrá visualizar una lista de las comunidades a las que pertenece
Autor	administrador
Pre-Condición	El usuario logueado como administrador El administrador debe pertenecer a las comunidades
Post-Condición	La lista de comunidades esta visualizada en la interfaz
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario pulsa el botón del menú 3. El usuario selecciona "Comunidades" 4. El sistema muestra una lista con las comunidades del usuario 	
Flujo alternativo	

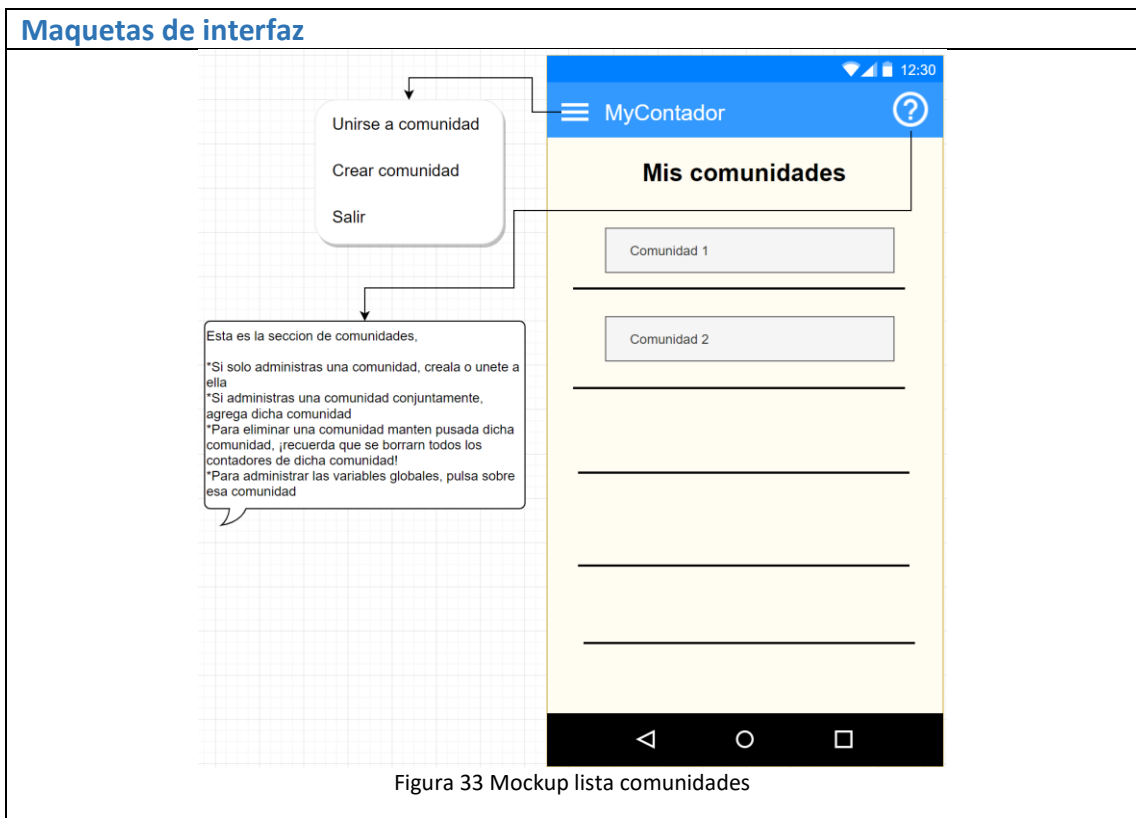


Tabla 22 CU Ver lista de comunidades

Nombre	Crear una comunidad
Descripción	El administrador podrá crear una nueva comunidad
Autor	administrador
Pre-Condición	El usuario logueado como administrador
Post-Condición	La nueva comunidad está en el sistema
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario abre la app 2. El usuario pulsa el botón del menú 3. El usuario selecciona "Comunidades" 4. El sistema muestra una lista con las comunidades del usuario 5. El usuario pulsa el botón del menú 6. El usuario selecciona Crear comunidad 7. El sistema muestra el formulario para añadir una comunidad 8. El usuario introduce el nombre 9. El usuario introduce la contraseña 10. El usuario pulsa sobre el botón "Crear" 11. El sistema crea la comunidad 12. El sistema vuelve a la pantalla de lista de comunidades 13. El sistema muestra la nueva comunidad en la lista de comunidades 	
Flujo alternativo	

1. El usuario abre la app
2. El usuario pulsa el botón del menú
3. El usuario selecciona "Comunidades"
4. El sistema muestra una lista con las comunidades del usuario
5. El usuario pulsa el botón del menú
6. El usuario selecciona "Crear comunidad"
7. El sistema muestra el formulario para añadir una comunidad
8. El usuario introduce el nombre
9. El usuario pulsa sobre el botón "Crear"
10. El sistema genera automáticamente una contraseña
11. El sistema crea la comunidad
12. El sistema vuelve a la pantalla de lista de comunidades

El sistema muestra la nueva comunidad en la lista de comunidades

Maquetas de interfaz



Figura 34 Mockup crear comunidad

Tabla 23 CU Crear una comunidad

Nombre	Unirse a una comunidad
Descripción	El administrador podrá unirse a una nueva comunidad
Autor	administrador
Pre-Condición	El usuario logueado como administrador El usuario está en la ventana de comunidades El usuario no pertenece a la comunidad
Post-Condición	El usuario pertenece a la comunidad
Flujo normal	
1. El usuario pulsa el botón del menú	

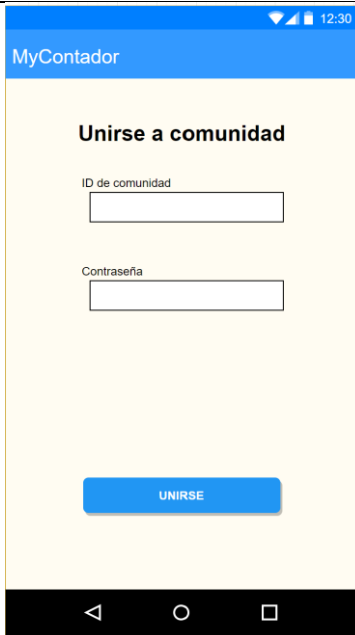
<ol style="list-style-type: none"> 2. El usuario selecciona "Unirse a una comunidad" 3. El sistema muestra el formulario para añadir una comunidad 4. El usuario introduce el id de la comunidad 5. El usuario introduce la contraseña 6. El usuario pulsa sobre el botón "Unirse" 7. El sistema lo une a la comunidad 8. El sistema vuelve a la pantalla de lista de comunidades 9. El sistema muestra la nueva comunidad en la lista de comunidades 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. El usuario pulsa el botón del menú 2. El usuario selecciona "Unirse a una comunidad" 3. El sistema muestra el formulario para añadir una comunidad 4. El usuario introduce el id de la comunidad 5. El usuario introduce la contraseña incorrecta 6. El usuario pulsa sobre el botón "Unirse" 7. El sistema muestra un mensaje indicando que la comunidad no existe 	
Maquetas de interfaz	
 <p>Figura 35 Mockup unirse a comunidad</p>	

Tabla 24 CU Unirse a una comunidad

Nombre	Eliminar una comunidad
Descripción	El administrador podrá eliminar una comunidad
Autor	administrador
Pre-Condición	El usuario logueado como administrador El usuario está en la ventana de comunidades El usuario es el creador de la comunidad
Post-Condición	La comunidad no está en el sistema

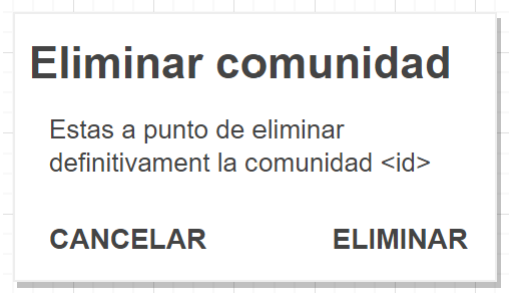
Flujo normal
<ol style="list-style-type: none"> 1. El usuario mantiene pulsada la comunidad 2. El sistema muestra un mensaje de confirmación para eliminarla 3. El usuario pulsa "Eliminar" 4. El sistema elimina la comunidad 5. La comunidad desaparece de la lista de comunidades
Flujo alternativo
<ol style="list-style-type: none"> 1. El usuario mantiene pulsada la comunidad 2. El sistema muestra un mensaje de confirmación para eliminarla 3. El usuario pulsa "Cancelar" 4. El sistema visualiza la lista de comunidades
Maquetas de interfaz
 <p>Figura 36 Mockup eliminar comunidad</p>

Tabla 25 CU Eliminar una comunidad

Nombre	Abandonar una comunidad
Descripción	El administrador podrá abandonar una comunidad
Autor	administrador
Pre-Condición	El usuario logueado como administrador El usuario está en la ventana de comunidades El usuario no es el creador de la comunidad
Post-Condición	La comunidad no contiene al administrador
Flujo normal	
	<ol style="list-style-type: none"> 6. El usuario mantiene pulsada la comunidad 7. El sistema muestra un mensaje de confirmación para salir de la comunidad 8. El usuario pulsa "Abandonar" 9. El sistema elimina al administrador elimina la comunidad 10. La comunidad desaparece de la lista de comunidades
Flujo alternativo	
	<ol style="list-style-type: none"> 5. El usuario mantiene pulsada la comunidad 6. El sistema muestra un mensaje de confirmación para salir de la comunidad 7. El usuario pulsa "Cancelar" 8. El sistema visualiza la lista de comunidades

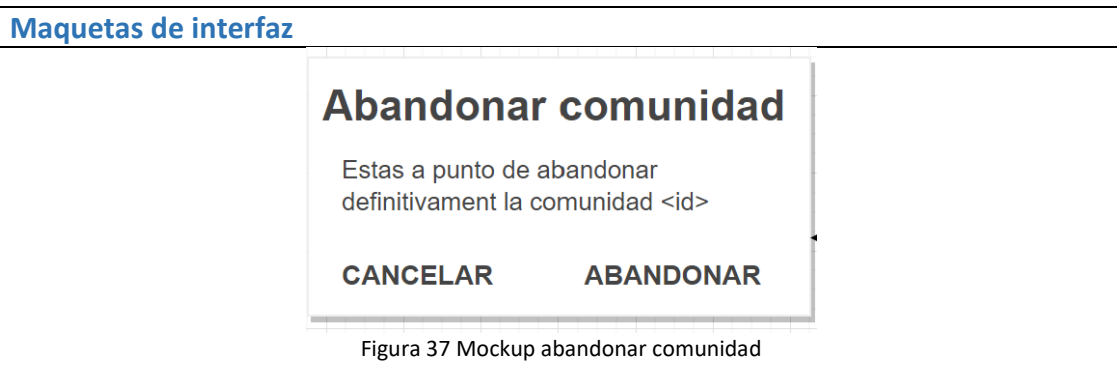


Tabla 26 CU Abandonar una comunidad

Nombre	Editar los parámetros de una comunidad
Descripción	El administrador podrá editar los parámetros una comunidad
Autor	administrador
Pre-Condición	El usuario logueado como administrador El usuario está en la ventana de comunidades El usuario pertenece a la comunidad
Post-Condición	El parámetro está modificado
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario selecciona la comunidad 2. El usuario selecciona el parámetro 3. El sistema muestra una ventana para introducir el parámetro 4. El usuario introduce el nuevo valor 5. El usuario pulsa el botón "aceptar" 6. El sistema actualiza el valor del parámetro 7. El nuevo valor se actualiza en la interfaz 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. El usuario selecciona la comunidad 2. El usuario selecciona el parámetro 3. El sistema muestra una ventana para introducir el parámetro 4. El usuario introduce el nuevo valor 5. El usuario pulsa el botón "Cancelar" 6. El sistema no actualiza el valor del parámetro 7. El antiguo valor se muestra en la interfaz 	

Maquetas de interfaz

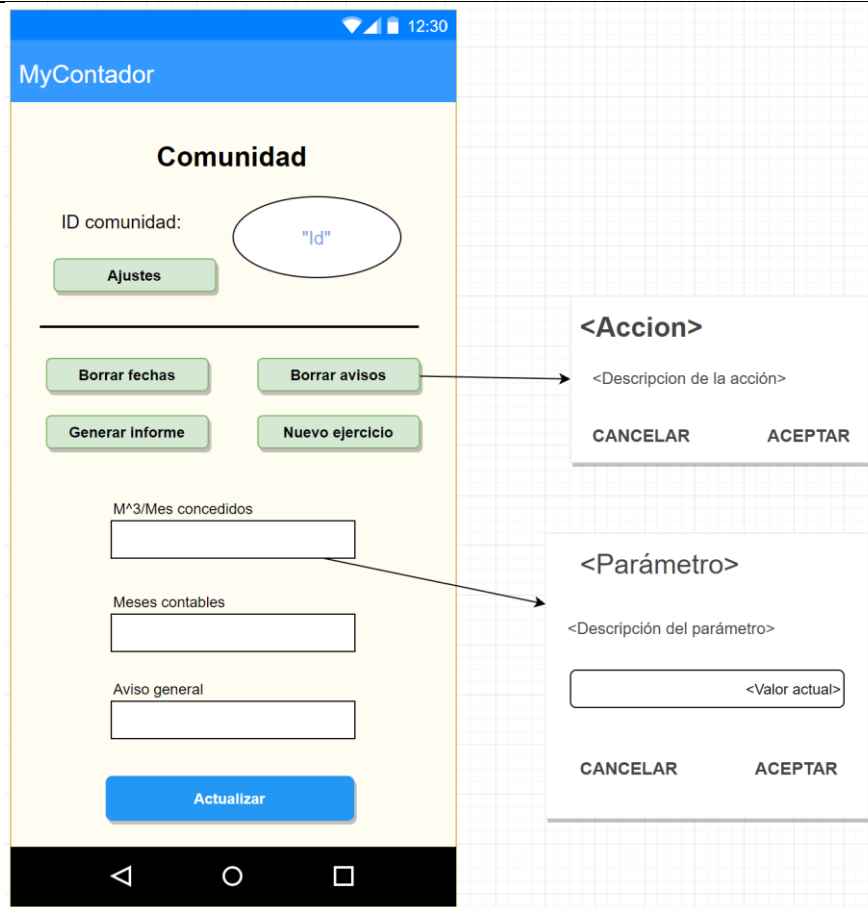


Figura 38 Mockup comunidad

Tabla 27 CU Editar los parámetros de una comunidad

Nombre	Editar los ajustes de una comunidad
Descripción	El administrador podrá editar los ajustes de una comunidad
Autor	administrador
Pre-Condición	El usuario logueado como administrador El usuario está en la ventana de comunidades El usuario pertenece a la comunidad
Post-Condición	El valor del ajuste está modificado
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario selecciona la comunidad 2. El usuario selecciona el botón "Ajustes" 3. El sistema muestra la ventana con los ajustes de la comunidad 4. El usuario introduce el nuevo valor 5. El usuario pulsa el botón "aceptar" 6. El sistema actualiza el valor del parámetro 7. El nuevo valor se actualiza en la interfaz 	
Flujo alternativo	

1. El usuario selecciona la comunidad
2. El usuario selecciona el botón "Ajustes"
3. El sistema muestra la ventana con los ajustes de la comunidad
4. El usuario introduce el nuevo valor
5. El usuario regresa a la ventana anterior
6. El sistema no actualiza el valor del parámetro

Maquetas de interfaz

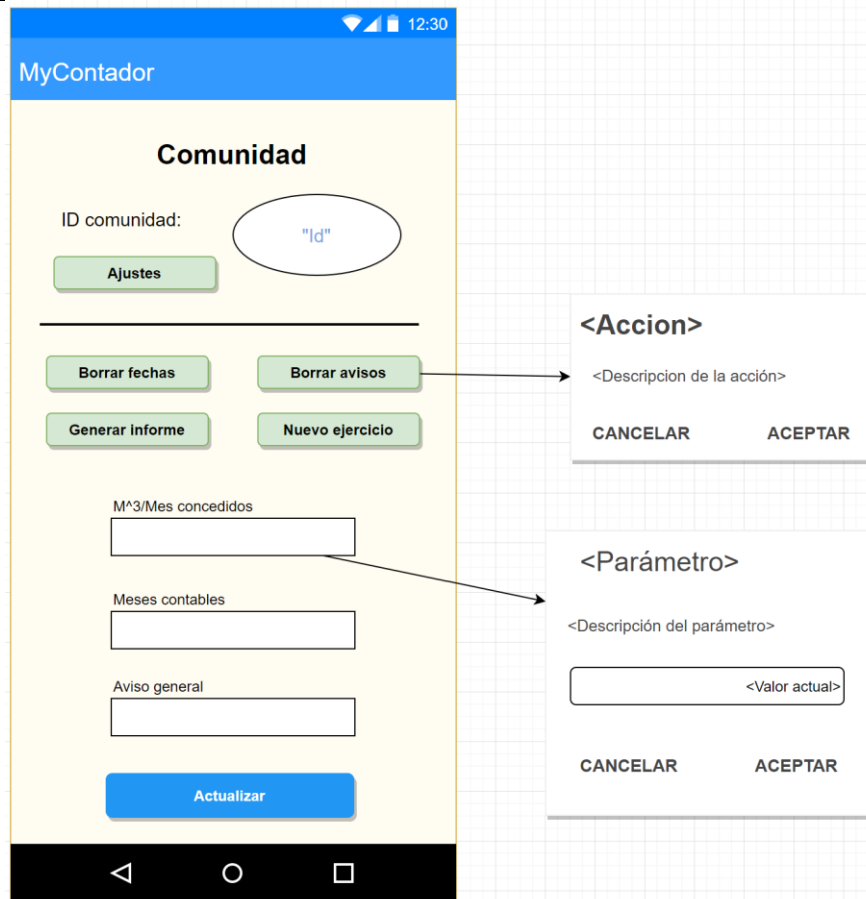


Figura 39 Mockup comunidad

Tabla 28 CU Editar los ajustes de una comunidad

Nombre	Eliminar las fechas de todos los contadores
Descripción	El administrador podrá eliminar a la vez las fechas de todos los contadores de una comunidad
Autor	administrador
Pre-Condición	El usuario logueado como administrador El usuario está en la ventana de comunidades El usuario pertenece a la comunidad
Post-Condición	No existe ningún contador en la comunidad con fecha de lectura
Flujo normal	
1. El usuario selecciona la comunidad	

2. El usuario selecciona el botón "Borrar fechas"
3. El sistema muestra una mensaje de advertencia indicando si desea realizar dicha acción
4. El usuario pulsa el botón "Ok"
5. El sistema elimina las fechas de todos los contadores
6. El sistema vuelve a la ventana anterior

Flujo alternativo

1. El usuario selecciona la comunidad
2. El usuario selecciona el botón "Borrar fechas"
3. El sistema muestra una mensaje de advertencia indicando si desea realizar dicha acción
4. El usuario pulsa el botón "Cancelar"
5. El sistema vuelve a la ventana anterior

Maquetas de interfaz

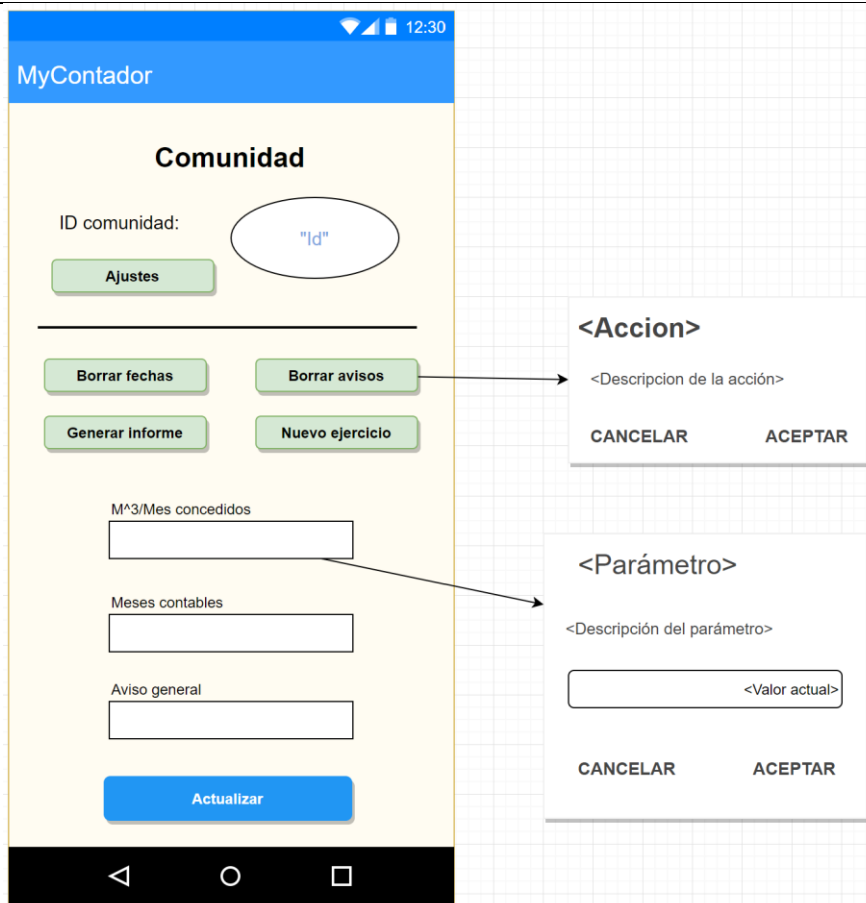


Figura 40 Mockup comunidad

Tabla 29 CU Eliminar las fechas de todos los contadores

Nombre	Eliminar los mensajes de todos los contadores
Descripción	El administrador podrá eliminar a la vez los mensajes de todos los contadores de una comunidad
Autor	administrador

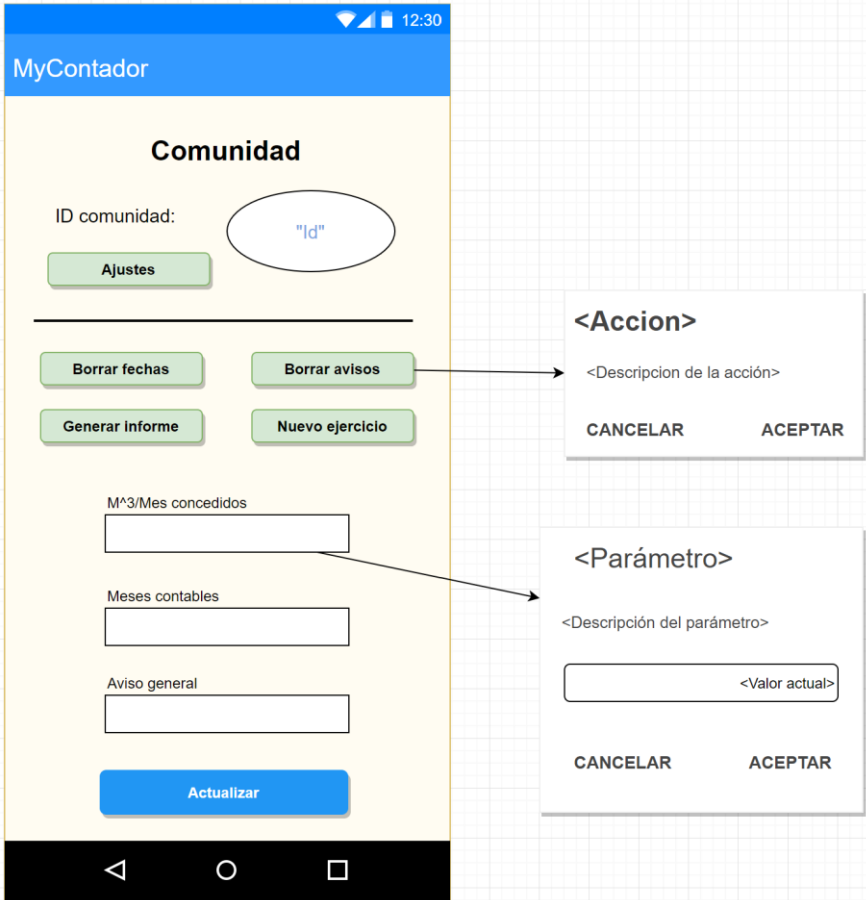
Pre-Condición	El usuario logueado como administrador El usuario está en la ventana de comunidades El usuario pertenece a la comunidad
Post-Condición	No existe ningún contador en la comunidad con un mensaje
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario selecciona la comunidad 2. El usuario selecciona el botón "Eliminar mensajes" 3. El sistema muestra una mensaje de advertencia indicando si desea realizar dicha acción 4. El usuario pulsa el botón "Ok" 5. El sistema elimina los mensajes de todos los contadores 6. El sistema vuelve a la ventana anterior 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. El usuario selecciona la comunidad 2. El usuario selecciona el botón ""Eliminar mensajes" 3. El sistema muestra una mensaje de advertencia indicando si desea realizar dicha acción 4. El usuario pulsa el botón "Cancelar" 5. El sistema vuelve a la ventana anterior 	
Maquetas de interfaz	
 <p>Figura 41 Mockup comunidad</p>	

Tabla 30 CU Eliminar los mensajes de todos los contadores

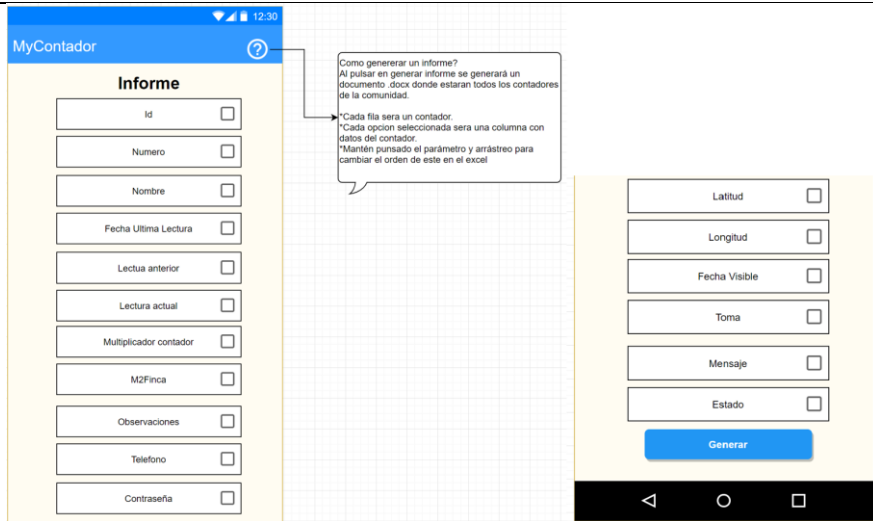
Nombre	Generar el informe de la comunidad
Descripción	El administrador de una comunidad podrá generar un informe con los datos de todos los contadores de una comunidad
Autor	administrador
Pre-Condición	El usuario logueado como administrador El usuario está en la ventana de comunidades El usuario pertenece a la comunidad
Post-Condición	El informe esta generado
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario selecciona la comunidad 2. El usuario selecciona el botón "Generar informe " 3. El sistema muestra una interfaz para seleccionar los datos a exportar 4. El usuario selecciona y ordena los datos a exportar 5. El usuario pulsa el botón "generar informe" 6. El sistema muestra un mensaje con el informe adjunto para que usuario lo guarde en su dispositivo 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. El usuario selecciona la comunidad 2. El usuario selecciona el botón "Generar informe " 3. El sistema muestra una interfaz para seleccionar los datos a exportar 4. El usuario selecciona y ordena los datos a exportar 5. El usuario pulsa el botón "generar informe" 6. El sistema muestra un mensaje de error indicado que no se ha seleccionado ningún campo 	
Maquetas de interfaz	
	

Figura 42 Mockup informe comunidad

Tabla 31 CU Generar el informe de la comunidad

Nombre	Desplazar la última lectura de todos los contadores
Descripción	El administrador podrá poner el valor de la última lectura en la lectura anterior y dejar la última lectura a 0 ala vez en todos los contadores
Autor	administrador
Pre-Condición	El usuario logueado como administrador EL usuario está en la ventana de comunidades El usuario pertenece a la comunidad
Post-Condición	Los contadores tienen la última lectura desplazada
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario selecciona la comunidad 2. El usuario selecciona el botón "Nuevo ejercicio" 3. El sistema muestra una mensaje de advertencia indicando si desea realizar dicha acción 4. El usuario pulsa el botón "Ok" 5. El sistema desplaza la ultima lectura de todos los contadores 6. El sistema vuelve a la ventana anterior 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. El usuario selecciona la comunidad 2. El usuario selecciona el botón "Nuevo ejercicio" 3. El sistema muestra una mensaje de advertencia indicando si desea realizar dicha acción 4. El usuario pulsa el botón "Cancelar" 5. El sistema vuelve a la ventana anterior 	

Maquetas de interfaz

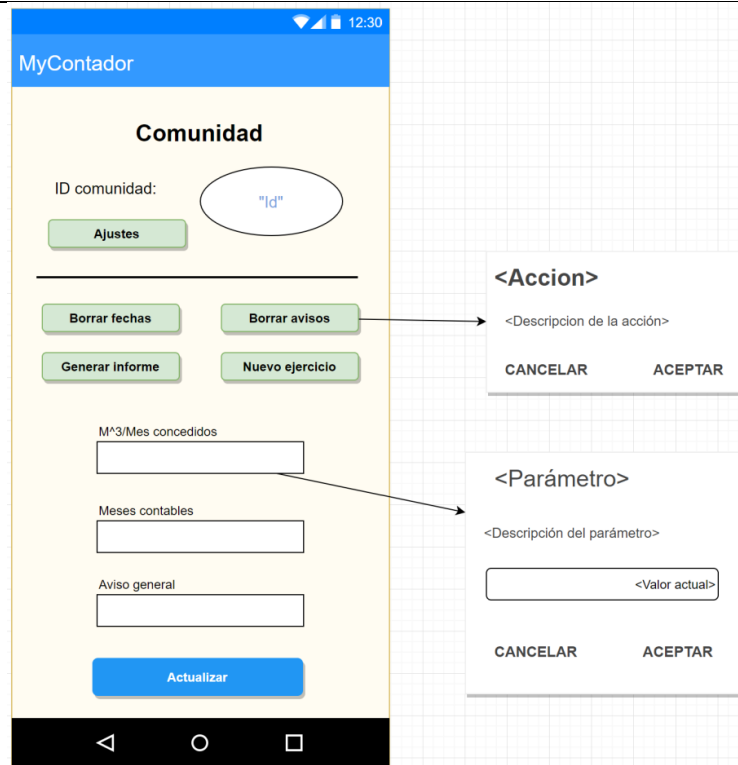


Figura 43 Mockup comunidad

Tabla 32 CU Desplazar la última lectura de todos los contadores

Nombre	Visualizar las estadísticas de los contadores
Descripción	El administrador podrá visualizar distintas estadísticas de los contadores
Autor	administrador
Pre-Condición	El usuario logueado como administrador El usuario está en la ventana de comunidades El usuario pertenece a la comunidad
Post-Condición	Las estadísticas se visualizan en pantalla
Flujo normal	
<ol style="list-style-type: none"> 1. El usuario selecciona la comunidad 2. El usuario selecciona el botón "Estadísticas" 3. El sistema muestra una ventana con las estadísticas de todos los contadores de la comunidad 	
Flujo alternativo	
<ol style="list-style-type: none"> 1. El usuario selecciona la comunidad 2. El usuario selecciona el botón "Estadísticas" 3. La comunidad no tiene contadores 4. El sistema indica en un mensaje que no existe ningún contador 	

Maquetas de interfaz

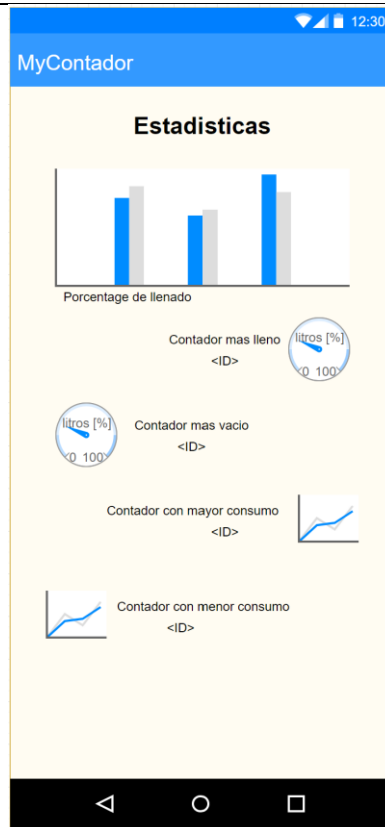


Figura 44 Mockup estadísticas

Tabla 33 CU Visualizar las estadísticas de los contadores

6

Metodología de desarrollo

6.1 Metodología

Para el desarrollo de la aplicación se ha optado por la metodología de **scrum**, mediante la cual se han ido implementado los distintos requisitos de la aplicación.

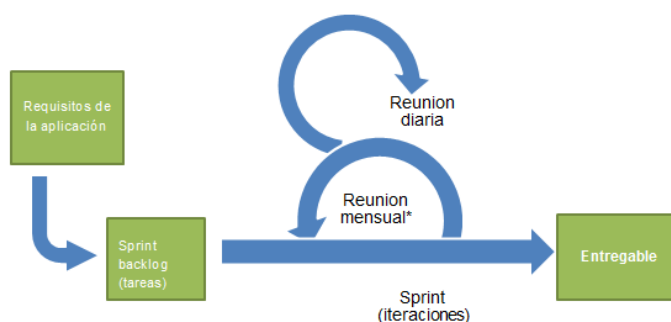


Figura 45 Metodología scrum

En Scrum se realizan entregas

parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

6.2 Entorno

El entorno de trabajo ha sido conjuntamente con el cliente. Las distintas versiones del proyecto se han desarrollado teniendo siempre el feedback del cliente en todo momento, estando disponible cuando hiciese falta. Puesto que se ha optado por una metodología ágil, las reuniones con el cliente han sido frecuentes, sobre todo, en las primeras fases del proyecto.

6.3 Proceso

Todos los **roles** principales como *Product Owner*, *ScrumMaster*, *Equipo de desarrollo* han recaído sobre el desarrollador de la aplicación, mientras que los *Stakeholders* ha sido el cliente para el que se ha realizado todo.

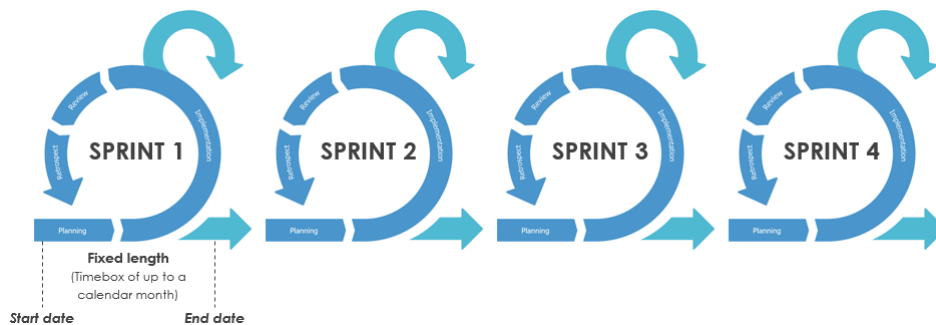


Figura 46 Proceso sprint

Se han omitido las **reuniones diarias**, ya que el equipo de desarrollo solo está formado por 1 persona, por lo que estas reuniones no son necesarias, en cambio, las **reuniones mensuales** sí que se han tenido en cuenta, ya que han sido donde se han definido los distintos sprints, aunque se han realizado de una forma periódica cuando se ha terminado el sprint en vez de mensualmente. Estas han sido las reuniones principales con el cliente para la evaluación de los requisitos, revisión de la aplicación y recolección de nuevos requisitos.

Como **Product backlog** se ha desarrollado en esta memoria, donde se han recogido todos los requisitos y características de la aplicación.

Para anotar todos los requisitos y tener un entorno para la organización del proyecto se ha utilizado Trello, tambien se ha usado para anotar todos los fallos que se han ido descubriendo durante el desarrollo de la aplicación.

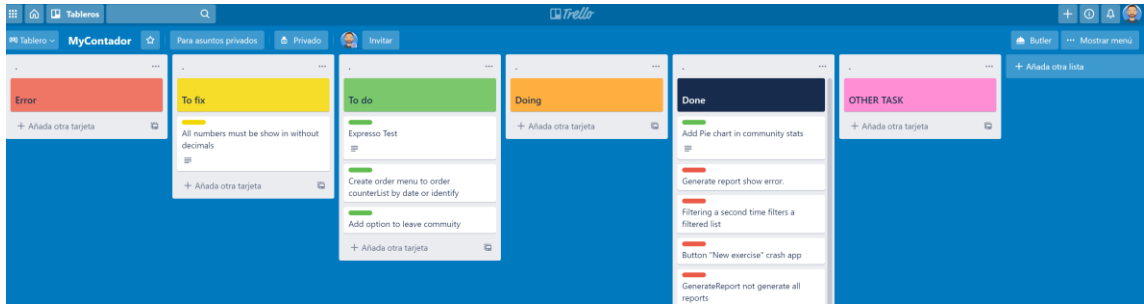


Figura 47 Tablón Trello

7

Bases de datos

7.1 SQL

Son todas aquellas bases de datos relaciones cuyas consultas son realizadas mediante el lenguaje de SQL, esta base de datos no se ha usado, debido a que nuestra aplicación utiliza la base de datos que proporciona Firebase. Aunque este tipo de base de datos es la más ideal para nuestra aplicación.

7.2 No SQL

Se trata de una base de datos cuya estructura no está organizada en tablas como la SQL.

Los datos pueden almacenarse en distintas estructuras, por ejemplo un JSON. Este es la base de datos que usa nuestra aplicación ya que es la que nos proporciona el servidor que estamos usando de Firebase.

7.3 Base de datos de la aplicación

En nuestro caso, usamos la base de datos Cloud Firestore, la cual nos permite tener los datos sincronizados de con el servidor. Esta es una base de datos flexible y escalable, por lo que es ideal para nuestra aplicación.

Esta base de datos está estructurada en Datos, Documentos y Colecciones.

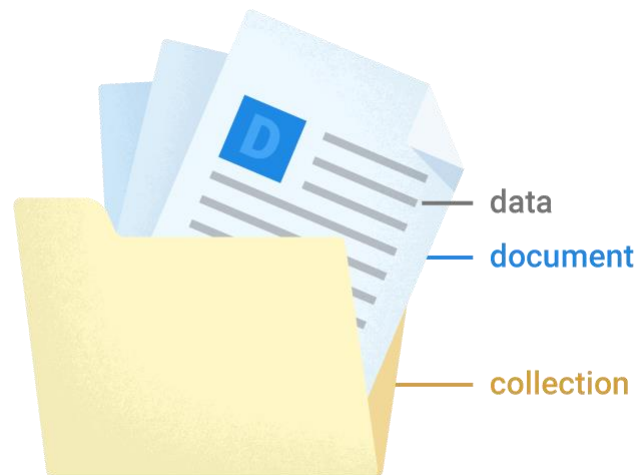


Figura 48 Estructura Cloud Firestore

Los datos se muestran como diccionarios clave-valor donde se pueden almacenar los distintos tipos:

- String: Para almacenar una secuencia de caracteres.
- Number: Para almacenar tanto números enteros como reales.
- Boolean: Para almacenar valores booleanos (true / false)
- Map: Para almacenar un diccionario.
- Array: Para almacenar una secuencia de valores de distintos tipos.
- Null: Para almacenar el valor null.
- Timestamp: Para almacenar las variables relacionadas con la fecha y el tiempo.
- Geopoint: Para almacenar unas coordenadas con una longitud y latitud.
- Reference: Para almacenar un puntero a otro documento.

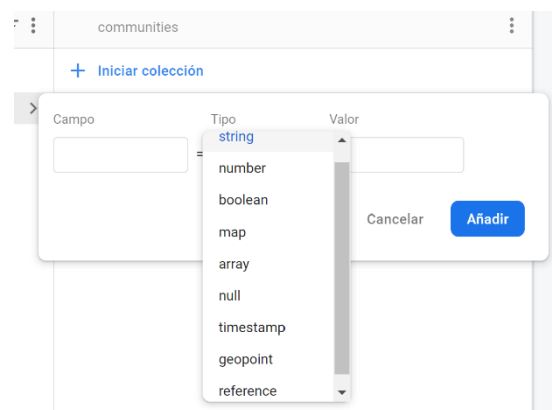


Figura 49 Tipos de datos Cloud Firestore

En Cloud Firestore se pueden añadir diferentes colecciones, las cuales apuntan a varios documentos los cuales deben tener un nombre distinto para poder diferenciarlos. Cada uno de estos documentos apuntan a una hoja de datos que es un diccionario de clave-valor en el que se pueden introducir los distintos datos mencionados anteriormente.

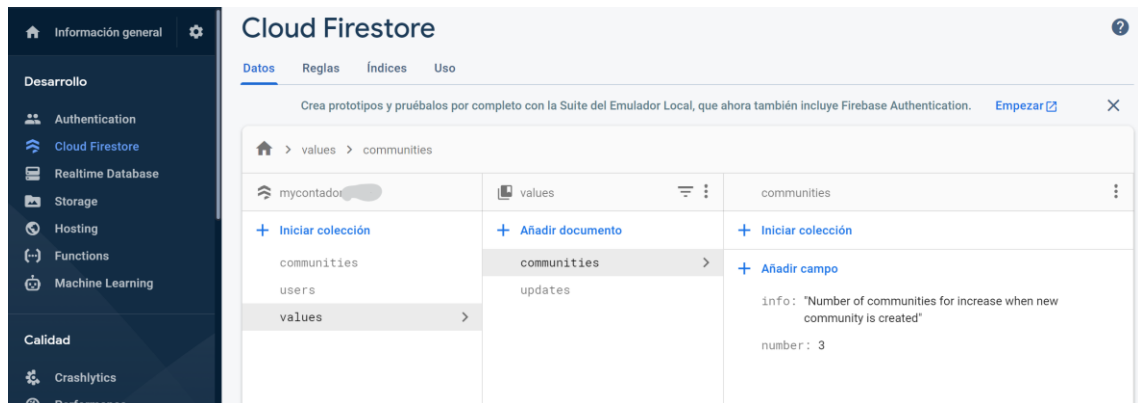


Figura 50 Consola de datos de Cloud Firestore

Los datos se introducen desde dentro de nuestra aplicación, aunque también puede usarse la consola de Firebase para poder visualizar o modificar la base de datos del proyecto que tengamos.

Un documento también puede apuntar a una colección, por lo que podríamos tener una colección "infinita" de colecciones -> documentos -> colecciones -> etc.

Esta funcionalidad puede ser muy útil para tener subcolecciones de documentos dentro de una colección.

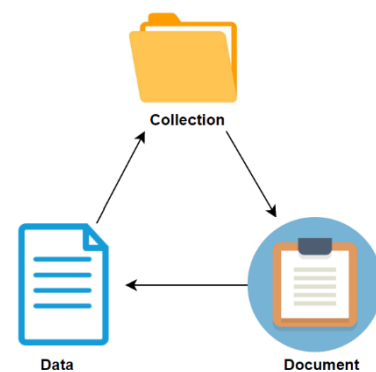


Figura 51 Modelo de datos Cloud Firestore

Cloud Firestore también nos proporciona otras funcionalidades como las **reglas** las cuales configuramos para dotar de una mayor seguridad a nuestro proyecto. También tenemos los **índices** los cuales podemos configurar para realizar consultas a la base de datos fácilmente y también tenemos un apartado **Uso** donde podemos observar el número de consultas hacia nuestra base de datos.

8

Diseño de la aplicación

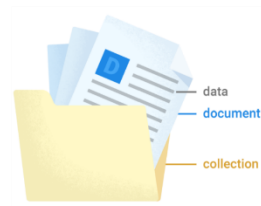
8.1 Diseño de la base de datos

Para la base de datos se ha utilizado el servicio de Cloud Firestore el cual nos proporciona un modelo de base de datos NOSQL, el cual ya hemos definido en esta memoria.

La base de datos se ha estructurado en 4 colecciones principales:

- Los **Usuarios** la cual contiene todos los usuarios de la aplicación, para cada usuario se guarda su correo, nombre, foto, número de contadores del modo usuario, valores de su último filtrado. También se guarda su uid el cual es único para cada usuario y permite identificar su cuenta de Google, permitiendo así que el usuario tenga la misma cuenta aunque cambie datos como por ejemplo, su correo.
- Las **Comunidades** las cuales contienen todas las comunidades identificadas por un valor numérico incremental, se ha realizado así ya que los usuarios tendrán que poner este número para unirse a la comunidad, por lo que se ha buscado la forma más simple posible. Cada comunidad contiene:
 - Una lista de **administradores** de dicha comunidad, la cual indica los usuarios que tienen acceso a dicha comunidad.

- Una lista de **valores** de la comunidad, los cuales son usados por los contadores para calcular distintos campos.
- Un **id** el cual es el identificador de la comunidad y permite que los administradores puedan unirse a esta.
- Un **nombre** de la comunidad.
- Un **dueño**, el cual apunta al uid del usuario que creo la comunidad.
- Una **contraseña** la cual es visible para todos los administradores de la comunidad.
- Una **lista de contadores**.
- Los **Valores** principales de la aplicación dentro del cual tenemos dos documentos
 - El **número de comunidades**, el cual usamos para averiguar de forma rápida cual es el numero de la última comunidad para la hora de añadir una nueva.
 - Las **actualizaciones** dentro de la cual se indica cuáles son las novedades de la actualización, la versión y la ruta donde descargar el fichero apk.



La base de datos de firebase se basa en el modelo noSQL de **colecciones->documentos->datos**

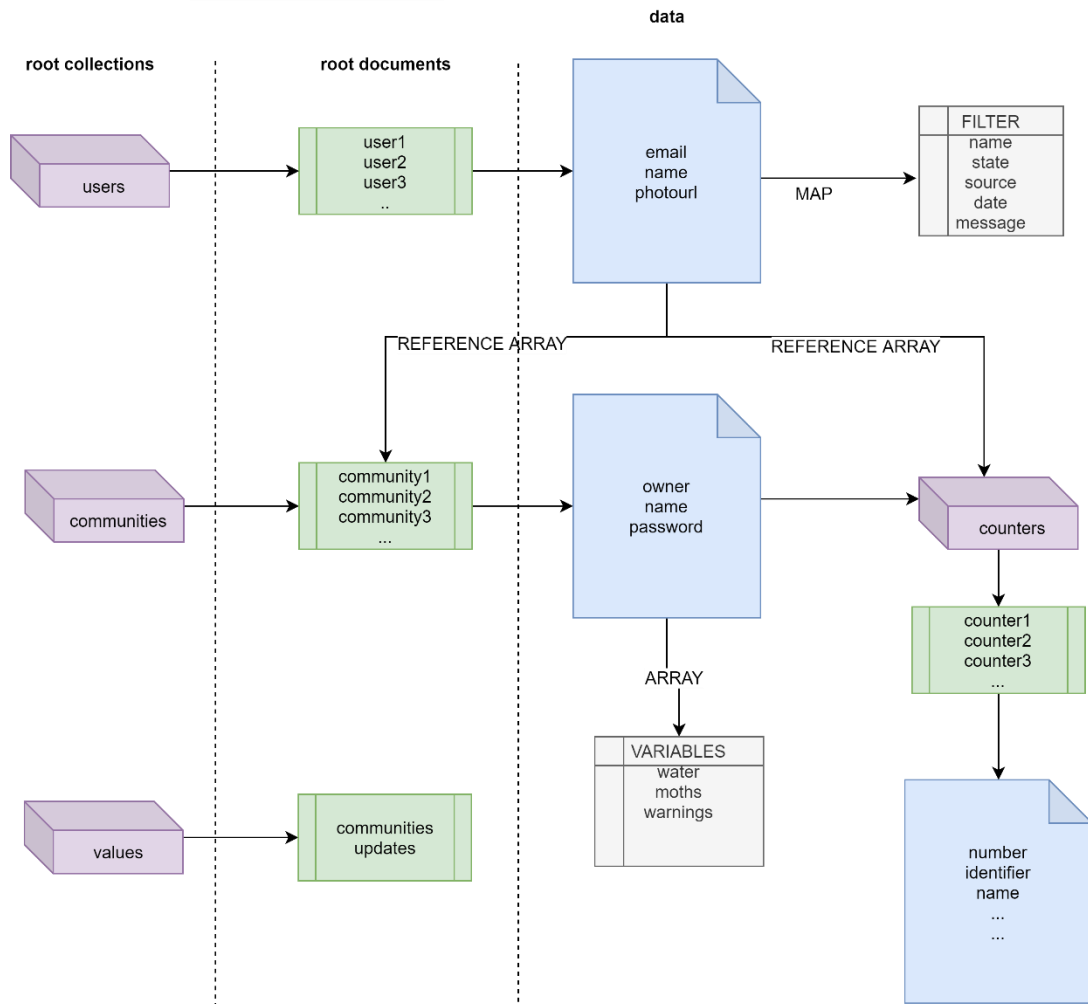


Figura 52 Diseño de la base de datos

8.1.1 Diagramas de la base de datos

La base de datos se ha estructurado de la anterior forma mencionada, para ello se ha basado en el siguiente diagrama, el cual se ha usado tanto para diseñar la base de datos como para crear las clases java que contienen dichas entidades

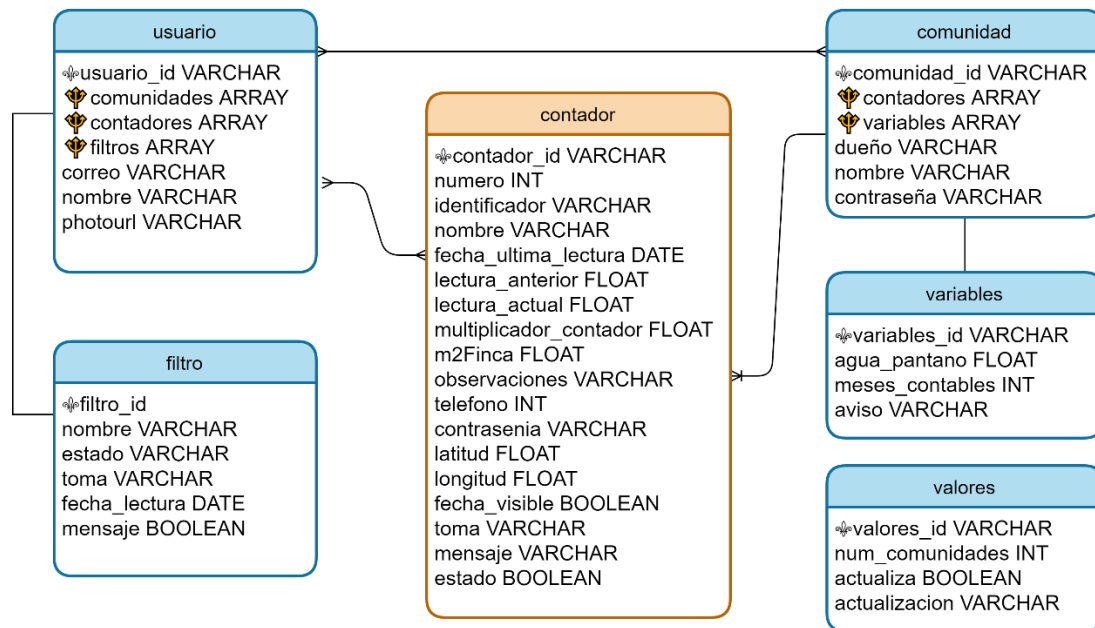


Figura 53 Diagrama de la base de datos

8.2 Diseño de la interfaz

Para estructurar la organización de las distintas ventanas de la aplicación se ha diseñado el siguiente diagrama el cual muestra la navegación entre las ventanas de la aplicación

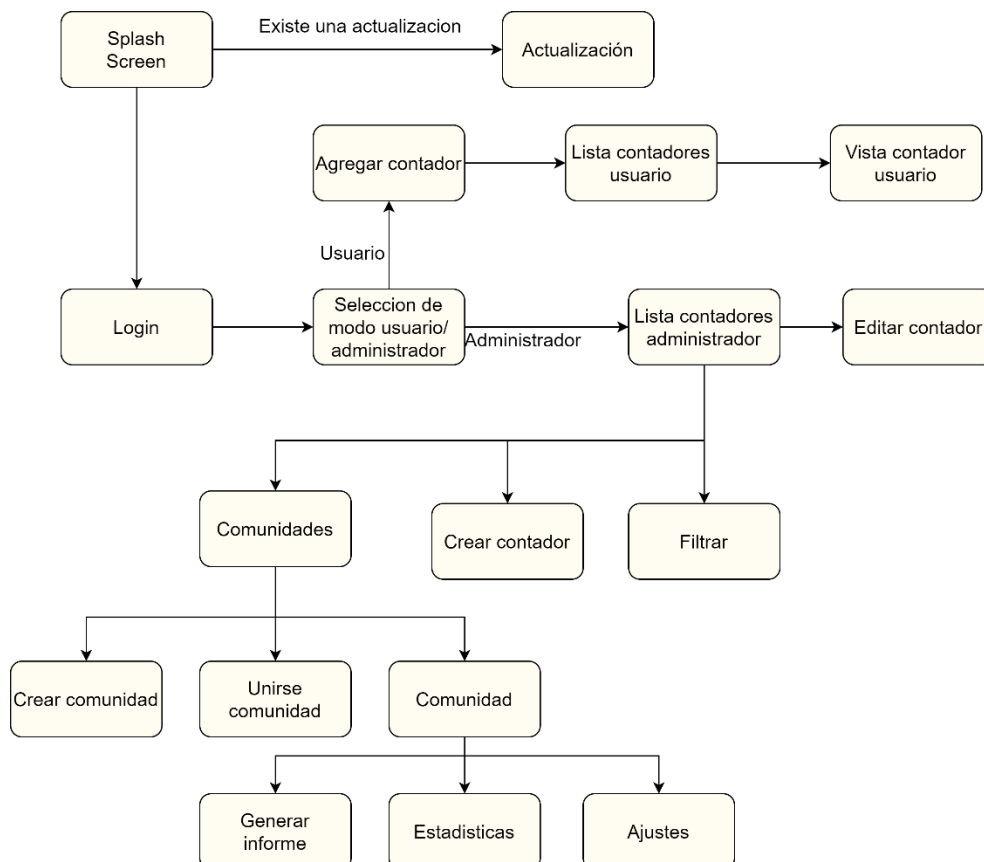


Figura 54 Diseño de la interfaz

8.3 Diseño del mockup

Durante la etapa de análisis de requisitos, se ha diseñado un mockup "interactivo" el cual muestra la navegación entre las distintas ventanas y la descripción del proyecto. Este mockup ha sido de vital importancia ya que ha sido de gran ayuda para acordar claramente las funcionalidades de la aplicación con el cliente.

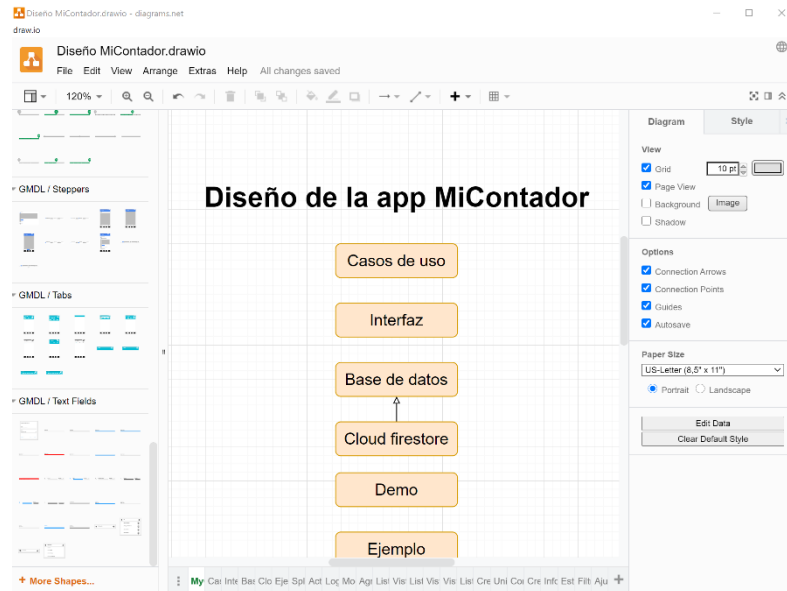


Figura 55 Diseño del mockup

Para ello se han creado una pestaña, cada una para cada ventana de la aplicación, se han enlazado entre ellas mediante enlaces y se ha exportado el fichero a un formato HTML el cual ha permitido la fácil interacción con el mockup

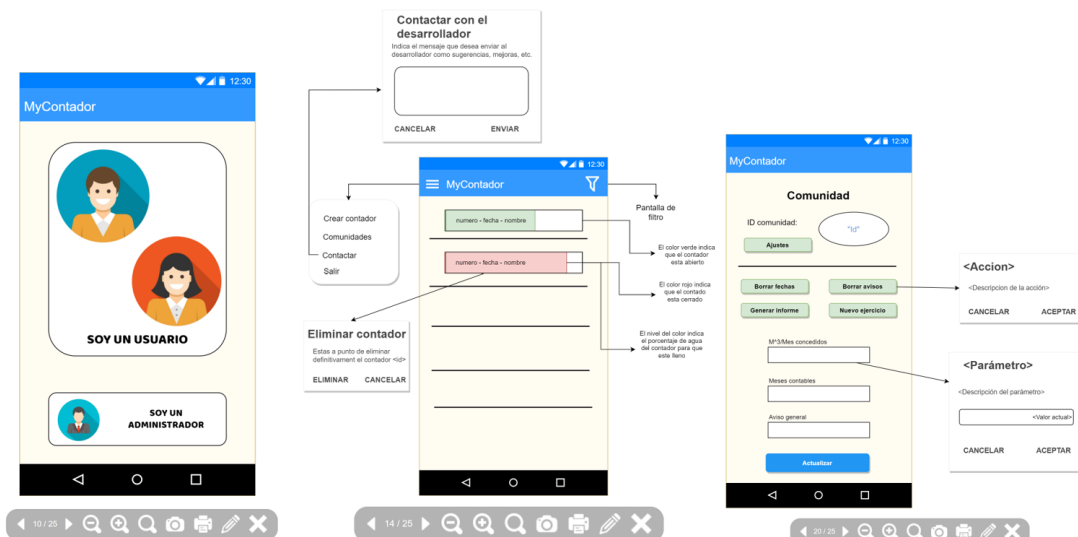


Figura 56 Uso del mockup

9

Desarrollo de la aplicación

9.1 Diagrama de clases

El diagrama de clases diseñado inicialmente es el siguiente

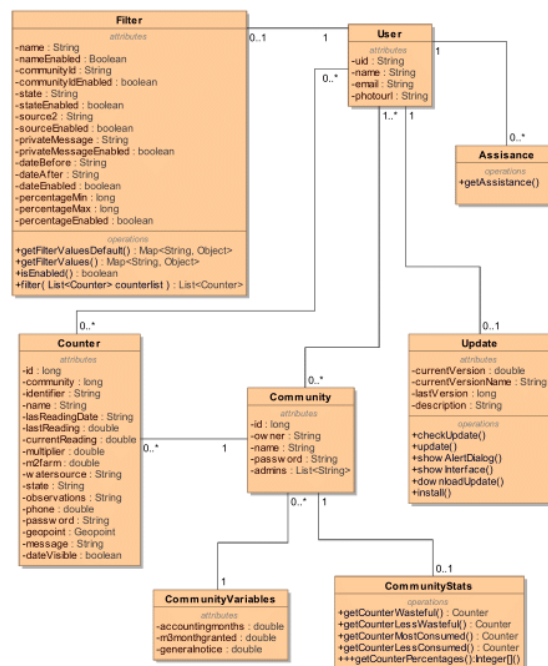


Figura 57 Diagrama de clases

También se ha obtenido el diagrama de paquetes de la aplicación a partir de Android Studio con el plugin **Code iris** la cual ha generado automáticamente el siguiente diagrama



Figura 58 Diagrama de paquetes

9.3 Estructura del proyecto

Para seguir la estructura de las aplicaciones android, se han agrupado las clases java desarrolladas en 4 paquetes:

- **Activities:** Contiene las clases asociadas a cada pantalla de la aplicación, sería el equivalente al controlador en el MVC.
- **Adapters:** Contiene las clases encargadas de crear los distintos componentes dinámicos que se visualiza en las pantallas de la aplicación.
- **Models:** Contiene las clases principales de la aplicación, asociadas al modelo en el MVC.
- **MainActivity.java:** Es la primera clase que se lanza en la aplicación al abrir esta. Su principal objetivo es el de comprobar si existe alguna actualización para la

app, y en caso contrario, lanzar la pantalla de inicio de sesión si el usuario no ha iniciado sesión o la pantalla del usuario o administrador en caso contrario.

Los ficheros de pruebas se almacenan en las carpetas test, e incluyen:

- IntegrationTest: Son los test orientados a la integración entre ventanas.
- UITest: Orientados a la interacción del usuario con la interfaz.
- UnitTest: Orientados al correcto funcionamiento individual de cada clase java.

Los ficheros de recursos usados por la aplicación están en la carpeta res, la cual tiene los siguientes módulos:

- Drawable: Contiene las distintas imágenes usadas por la aplicación, así como los iconos usados en las ventanas de esta y las estructuras de los botones y las barras de progresos.
- Font: Contiene las fuentes de los distintos tipos de letras usadas en la app.
- Layout: Contiene la estructura de todas las interfaces de la aplicación, asociadas a la vista en el MVC.
- Menu: Contiene la estructura de todas los menús de la aplicación, asociadas a la vista en el MVC.
- Mipmap: Contiene distintas imágenes asociadas al icono de la aplicación para que este tenga mas o menos resolución en función al dispositivo android en el que se ejecute la app.
- Values: Contiene varios datos como los colores usados en la app, el splashScreen, y los valores asociados a la traducción de la app.
- XML: Contiene una serie de valores necesarios para que la app sea capaz de descargar el fichero .apk de actualización y poder así actualizarse automáticamente.

10

Seguridad

10.1 Seguridad de la app



Figura 59 Archivo google-services.json

La aplicación sigue un diseño cliente-servidor, por lo que el sector crítico está en acceso al servidor para la obtención de datos, la vinculación entre la aplicación Android y el servidor de Firebase se realiza mediante un archivo en formato JSON el cual se

descarga desde la consola de Firebase asociada al proyecto, este contiene los datos necesarios para efectuar la conexión con el servidor de manera segura e identificada, de esta forma, solo nuestra aplicación puede conectarse con el servidor por lo que es aquí donde esta nuestro primer cortafuegos.

Una vez tenemos nuestra app vinculada con el servidor de Firebase, podemos configurar los accesos de la app a esta para asegurarnos que los usuarios acceden solo a los datos que les son permitido, esto se realiza mediante Firebase Rules.

10.2 Firebase Rules

El acceso a los datos se controla mediante Firebase Rules, permitiendo así proteger los datos para que estos solo sean accesibles con los permisos debidos para los usuarios legítimos.

De esta forma conseguimos proteger los datos de la siguiente forma:

- Los la aplicación solo puede acceder a los datos del servidor si previamente el usuario ha iniciado sesión con su cuenta de Google.
- Solo los usuarios registrados en la aplicación pueden acceder a los datos.
- Solo se permite la creación de nuevos formularios, evitando así que un usuario pueda modificar o leer algún formulario ajeno.
- Solo se permite la lectura de algunos datos como la ruta del apk de actualización, evitando así que cualquier usuario puede modificar dichos valores.

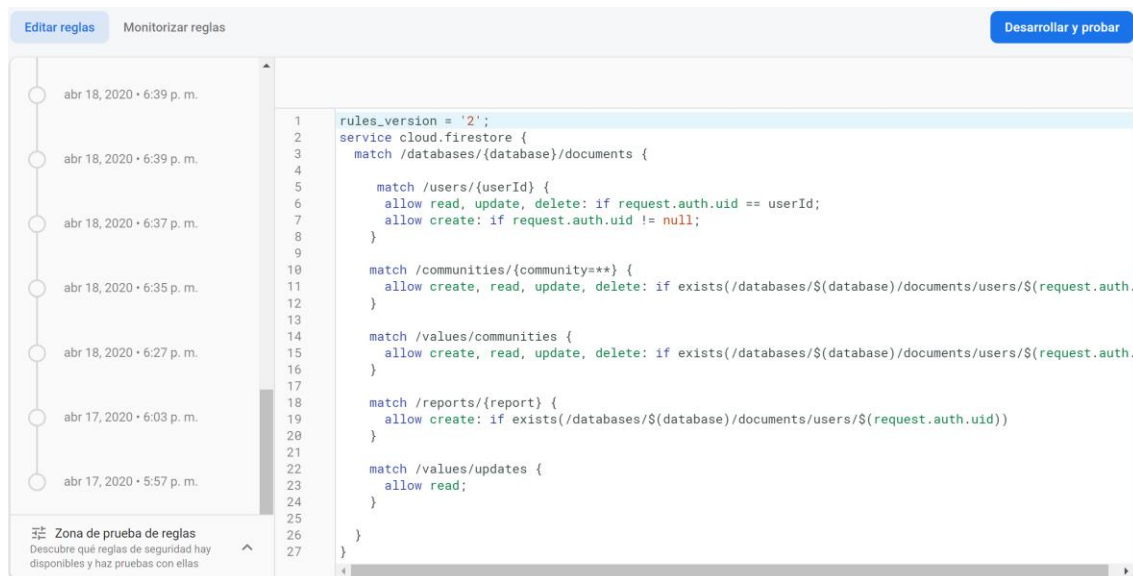


Figura 60 Firebase rules

10.3 Almacenamiento interno

La mayor parte de los datos se guarda en el servidor de Firebase, pero existen datos que se guardan en el dispositivo del móvil, como la caché para recordar cual fue el último perfil (usuario o administrador) seleccionado por el usuario, para ello utilizamos **SharedPreferences** el cual es un método que nos proporciona Android para guardar los datos de la app y que solo sean accesibles desde esta.

11

Pruebas

11.1 Descripción

Las pruebas es una de las fases mas importante del desarrollo del software, es de vital importancia ya que el software maneja datos privados de usuarios y se despliega sobre un entorno real, por lo que es de vital importancia que todo funcione lo mejor posible. Es imposible tener un software 100% libre de fallos, ya que aunque se controle al máximo la calidad del software desarrollado, este usa software ajeno que podría contener igualmente fallos, aun así se han intentado abarcar el mayor número de casos de prueba, para ello se han desarrollado diferentes pruebas en distintos entornos.

11.2 Emulador

Android Studio nos proporciona un emulador para probar la aplicación. Esta es la forma mas directa de probar el software, ya que es la que mas integrada esta con el entorno de desarrollo, aunque no ha sido la principal.

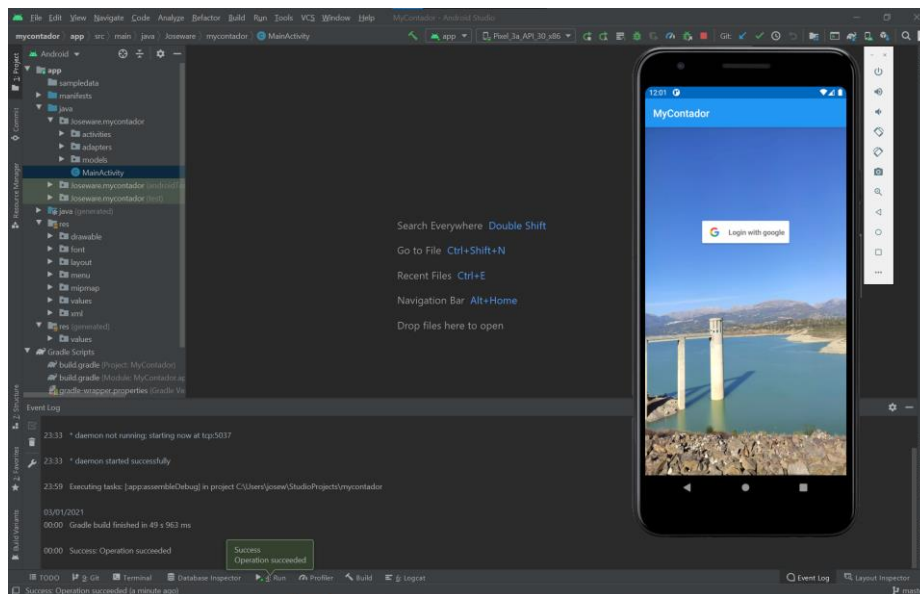


Figura 61 Emulador Android Studio

11.3 ADB

ADB es el protocolo de conexión con un dispositivo android, para usarlo hay que conectar un dispositivo físico a Android Studio mediante un cable usb o wifi, posteriormente lanzamos la aplicación desde Android Studio, visualizándose en el dispositivo y teniendo en todo momento los datos de lo que esta pasando en la consola, como los errores, mensajes de información, etc.

Dada que es la forma más parecida al entorno real, ha sido la principal forma de desarrollar las pruebas. Para cada nueva característica añadida, se ha lanzado la aplicación en el dispositivo físico mediante ADB para poder probarla.

11.4 Espresso

Espresso (Wikipedia, s.f.) es un framework de testing open source lanzado por Google el cual provee una API que permite crear pruebas de interfaz de usuario (UI) para simular interacciones de usuarios en una aplicación Android. Es una buena práctica simular los diferentes escenarios en los que el usuario puede interactuar con una aplicación para evitar que este se encuentre con resultados inesperados o bien tenga una mala experiencia al momento de su uso. Es por esta razón que es recomendable la creación de un entorno de pruebas vinculadas a la UI con el fin de asegurarnos que la aplicación está funcionando correctamente.

11.4.1 Pruebas de unidad

Son las encargadas de comprobar el correcto funcionamiento de una unidad de código.

Dichas pruebas están enfocadas a asegurar que las funciones, parámetros, etc. de cada clase java cumple con lo especificado.

Las pruebas de unidad están implementadas en el fichero **UnitTest**.

11.4.2 Pruebas de Integración

Son las encargadas de asegurar la correcta interacción entre los distintos módulos de la aplicación, es decir, que las llamadas entre módulos y los datos que se envíen entre ellos se haga de forma correcta y sin errores.

Las pruebas de integración están implementadas en el fichero **IntegrationTest**.

11.4.3 Pruebas de Extremo a extremo

Son las pruebas sobre las interfaces. Validan la aplicación a nivel de interfaz, simulando la interacción del usuario con esta y verificando el correcto funcionamiento a través de las distintas interfaces.

Las pruebas de extremo a extremo están implementadas en el fichero **UITest**.

11.5 Firebase Crashlytics

Es una herramienta para la provisión de informes en tiempo real, la cual está integrada con la aplicación para que cuando un usuario provoque algún fallo en la app, este pueda enviarse al desarrollador con datos suficientes para que este pueda resolver el problema.

Esta es la forma más importante de realización de las pruebas, ya que es la que permite corregir y abarcar el mayor número de fallos de la aplicación.

11.6 Firebase Test Lab

Es un servicio proporcionado por firebase, el cual nos permite hacer pruebas automáticas en múltiples dispositivos. Para ello tan solo tenemos que subir el fichero APK de nuestra aplicación y Test Lab se encarga del resto. Estas pruebas se han realizado para asegurar la compatibilidad con múltiples dispositivos.

Si la prueba detecta algún error, entonces Test Lab nos proporciona todos los datos necesarios para solucionar el error, si por el contrario todo funciona, obtenemos un diagrama en el cual podemos observar el recorrido de la prueba realizada.

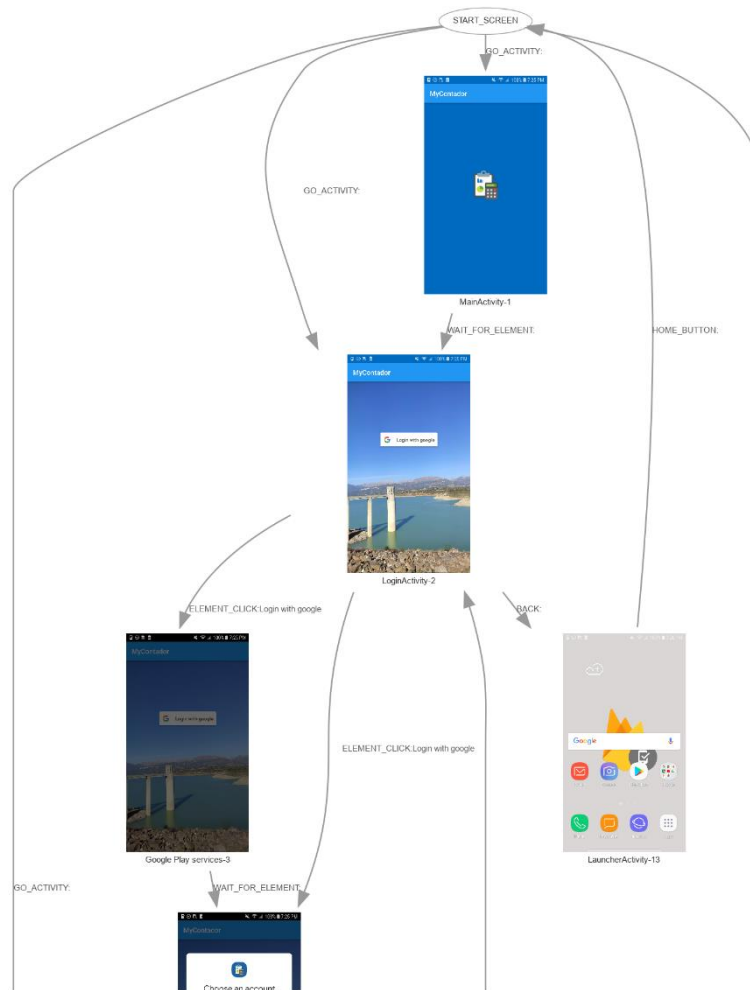


Figura 62 Diagrama de Test Lab

11.7 Pruebas realizadas

La aplicación ha sido testeada para satisfacer todos los casos de uso, además de ello, también ha sido testeada con los siguientes casos de prueba a nivel de UI.

Código	TST1
Acción	Iniciar sesión con diferentes cuentas de google
Salida esperada	
Que los datos de cada cuenta sean distintos	
Salida obtenida	
Los datos son distintos	

Tabla 34 Caso de prueba TST1

Código	TST2
Acción	Iniciar sesión sin conexión
Salida esperada	
No se inicia sesión y salta un mensaje de error de conexión	
Salida obtenida	
No se inicia sesión	

Tabla 35 Caso de prueba TST2

Código	TST3
Acción	Ver la lista de contadores cuando no hay internet
Salida esperada	
La lista de contadores debe visualizarse	
Salida obtenida	
Se visualiza la lista de contadores	

Tabla 36 Caso de prueba TST3

Código	TST4
Acción	Añadir un contador introduciendo datos válidos
Salida esperada	
El nuevo contador debe visualizarse en la lista de contadores	
Salida obtenida	
El nuevo contador se visualiza en la lista de contadores	

Tabla 37 Caso de prueba TST4

Código	TST5
Acción	Añadir un contador introduciendo datos inválidos
Salida esperada	
Debe salir un mensaje indicando un error	
Salida obtenida	
Sale un mensaje indicando que el contador no existe	

Tabla 38 Caso de prueba TST5

Código	TST6
Acción	Pulsar sobre un contador
Salida esperada	
El identificador del contador debe ser el mismo que el del contador seleccionado en la lista	
Salida obtenida	
El identificador del ítem de la lista pulsado es el mismo que el del contador	

Tabla 39 Caso de prueba TST6

Código	TST7
Acción	Pulsar sobre un campo de un contador en modo usuario
Salida esperada	
Debe salir un mensaje de descripción	
Salida obtenida	
Aparece un mensaje indicando la descripción del campo seleccionado	

Tabla 40 Caso de prueba TST7

Código	TST8
Acción	Eliminar un contador en modo usuario
Salida esperada	
El contador debe desaparecer de la lista y de la base de datos	
Salida obtenida	
El contador desaparece de la lista y de la base de datos	

Tabla 41 Caso de prueba TST8

Código	TST9
Acción	Crear un contador introduciendo solo el identificador

Salida esperada
El nuevo contador debe aparecer en la lista de contadores
Salida obtenida
El nuevo contador aparece en la lista de contadores

Tabla 42 Caso de prueba TST9

Código	TST10
Acción	Eliminar un contador en modo administrador
Salida esperada	
El contador debe eliminarse para todos los usuarios y Administradores que lo tengan	
Salida obtenida	
El contador se elimina de todas las listas de usuarios y administradores	

Tabla 43 Caso de prueba TST10

Código	TST11
Acción	Cambiar cualquier dato de un contador
Salida esperada	
El dato debe actualizarse para todos los usuarios y administradores del contador	
Salida obtenida	
El dato se actualiza para todos los usuarios y administradores del contador	

Tabla 44 Caso de prueba TST11

Código	TST12
Acción	Deshabilitar ocultar la fecha de un usuario
Salida esperada	
Los usuarios que tengan el contador no deben visualizar la fecha	
Salida obtenida	
Los usuarios que tengan el contador no visualizan la fecha	

Tabla 45 Caso de prueba TST12

Código	TST13
Acción	Abrir la localización de un contador
Salida esperada	
Debe abrirse google maps con la localización en el mapa	
Salida obtenida	

Se visualiza en google maps la localización del contador
--

Tabla 46 Caso de prueba TST13

Código	TST14
Acción	Eliminar una comunidad
Salida esperada	
La comunidad debe desaparecer de la listas de los administradores que la tengan y todos los contadores de esta deben desaparecer de la lista de todos los administradores y usuarios que la tengan	
Salida obtenida	
La comunidad y todos los contadores desaparecen de todos los usuarios y administradores	

Tabla 47 Caso de prueba TST14

Código	TST15
Acción	Crear una comunidad
Salida esperada	
La comunidad debe aparecer en la lista de comunidades	
Salida obtenida	
La comunidad aparece en la lista de comunidades	

Tabla 48 Caso de prueba TST15

Código	TST16
Acción	Unirse a una comunidad
Salida esperada	
La comunidad debe aparecer en la lista de comunidades	
Salida obtenida	
La comunidad aparece en la lista de comunidades	

Tabla 49 Caso de prueba TST16

Código	TST17
Acción	Cambiar el nombre de una comunidad
Salida esperada	
El nuevo nombre de la comunidad debe aparecer para todos los administradores que estén en ella.	
Salida obtenida	

El nuevo nombre de la comunidad aparece para todos los administradores que están en ella.

Tabla 50 Caso de prueba TST17

Código	TST18
Acción	Cambiar la contraseña de una comunidad
Salida esperada	
La contraseña debe actualizarse y los administradores que estaban en esa comunidad deben seguir en ella	
Salida obtenida	
La contraseña se actualiza y los administradores que estaban en esa comunidad siguen en ella	

Tabla 51 Caso de prueba TST18

Código	TST19
Acción	Eliminar todas las fechas
Salida esperada	
Todos los contadores de la comunidad deben tener la fecha en blanco	
Salida obtenida	
Todos los contadores de la comunidad tienen la fecha en blanco	

Tabla 52 Caso de prueba TST19

Código	TST20
Acción	Eliminar todas los mensajes
Salida esperada	
Todos los contadores de la comunidad deben tener el mensaje privado en blanco	
Salida obtenida	
Todos los contadores de la comunidad tienen el mensaje privado en blanco	

Tabla 53 Caso de prueba TST20

Código	TST21
Acción	Generar un informe de todos los contadores
Salida esperada	
Debe generarse un fichero Excel que se guarda en el dispositivo con los datos de todos los contadores de la comunidad	
Salida obtenida	

Se genera un fichero Excel que se guarda en el dispositivo con los datos de todos los contadores de la comunidad

Tabla 54 Caso de prueba TST21

Código	TST22
Acción	seleccionar "Nuevo ejercicio"
Salida esperada	
Para todos los contadores, el valor de las lecturas anteriores debe ser el de la lectura actual	
Salida obtenida	
Para todos los contadores, el valor de las lecturas anteriores es el de la lectura actual	

Tabla 55 Caso de prueba TST22

Código	TST23
Acción	Visualizar las estadísticas
Salida esperada	
Los valores mostrados en las estadísticas deben corresponder con los contadores de la comunidad	
Salida obtenida	
Los valores mostrados en las estadísticas corresponden con los contadores de la comunidad	

Tabla 56 Caso de prueba TST23

Código	TST24
Acción	Cambiar los metros cúbicos concedidos en la comunidad
Salida esperada	
Todos los contadores de la comunidad deben actualizar sus valores que requieran de dicho dato.	
Salida obtenida	
Todos los contadores de la comunidad actualizan los valores que requieren de dicho dato.	

Tabla 57 Caso de prueba TST24

Código	TST25
Acción	Escribir una noticia general en la comunidad
Salida esperada	

Todos los usuarios que tengan añadido algún contador de la comunidad deben visualizar el mensaje al abrir la app
Salida obtenida
Todos los usuarios que tienen añadido algún contador de la comunidad visualizan el mensaje al abrir la app

Tabla 58 Caso de prueba TST25

Código	TST26
Acción	Escribir un mensaje privado en un contador
Salida esperada	
Todos los usuarios que tengan añadido el contador deben visualizar el mensaje al abrir la app	
Salida obtenida	
Todos los usuarios que tengan añadido el contador visualizan el mensaje al abrir la app	

Tabla 59 Caso de prueba TST26

Código	TST27
Acción	Filtrar por nombre
Salida esperada	
Solo deben aparecer en la lista los contadores que contengan la cadena introducida	
Salida obtenida	
Solo aparecen en la lista los contadores que contengan la cadena introducida	

Tabla 60 Caso de prueba TST27

Código	TST28
Acción	Filtrar por comunidad
Salida esperada	
Solo deben aparecer en la lista los contadores que pertenezcan a la comunidad introducida	
Salida obtenida	
Solo aparecen en la lista los contadores que pertenezcan a la comunidad introducida	

Tabla 61 Caso de prueba TST28

Código	TST29
Acción	Filtrar por fuente de agua
Salida esperada	

Solo deben aparecer en la lista los contadores que contengan la cadena introducida
Salida obtenida
Solo aparecen en la lista los contadores que contengan la cadena introducida

Tabla 62 Caso de prueba TST29

Código	TST30
Acción	Filtrar por mensaje privado
Salida esperada	
Solo deben aparecer en la lista los contadores que contengan la cadena introducida	
Salida obtenida	
Solo aparecen en la lista los contadores que contengan la cadena introducida	

Tabla 63 Caso de prueba TST30

Código	TST30
Acción	Filtrar estado
Salida esperada	
Solo deben aparecer en la lista los contadores que contengan la cadena introducida	
Salida obtenida	
Solo aparecen en la lista los contadores que contengan la cadena introducida	

Tabla 64 Caso de prueba TST31

Código	TST30
Acción	Filtrar por fecha
Salida esperada	
Solo deben aparecer en la lista los contadores aquellos cuya fecha de ultima lectura este dentro del intervalo introducido	
Salida obtenida	
Solo aparecen en la lista los contadores aquellos cuya fecha de ultima lectura está dentro del intervalo introducido	

Tabla 65 Caso de prueba TST32

Código	TST30
Acción	Filtrar por porcentaje
Salida esperada	
Solo deben aparecer en la lista los contadores aquellos cuyo porcentaje este dentro del intervalo introducido	

Salida obtenida
Solo aparecen en la lista los contadores aquellos cuyo porcentaje está dentro del intervalo introducido

Tabla 66 Caso de prueba TST33

12

Conclusiones

12.1 Conclusiones

Tras la realización del proyecto, se han remarcado la importancia del mantenimiento y de crear una aplicación fácilmente mantenible, ya que la mayor parte del tiempo es dedicada a la implementación de pequeñas características en el código que aunque no implemente ninguna funcionalidad, es de vital importancia para el desarrollo del mismo. También se le ha dado importancia a especificar del primer momento todos los requisitos, ya que el cliente tiende a pedir más de lo que se acordó inicialmente. También es una etapa muy importante el desarrollo del mockup, ya que es en este principalmente donde el cliente y el desarrollador pueden ponerse de acuerdo para acordar claramente cuales van a ser las características principales de la aplicación a desarrollar.

12.2 Líneas futuras

La principal línea futura es el mantenimiento de la aplicación, ya que esta depende de los servicios de Firebase y del sistema operativo de Android, por lo que si hubiese algún cambio en estos, habría que adaptar la aplicación para que fuese compatible.

También hacer mención a la creación de nuevos parámetros en los contadores, ya que es un sector en el que dependiendo del cliente, este podría añadir nuevos requisitos que necesitare.

También hacer mención a la continua corrección de errores, ya que, aunque se han controlado el máximo número de errores la aplicación puede tener algún fallo aún no observado, los cuales por ejemplo, puede ser alguna incompatibilidad con algún nuevo modelo de teléfono debido al gran ecosistema Android.

Como línea futura principal sería interesante la implementación de un sistema de Arduino/Raspberry pi que mediante el uso de un sensor conectado al contador y el uso de internet, fuese capaz de enviar los datos del contador a la base de datos de la nube, simplificando así la tarea del revisor del contador.

Bibliografía

Palomo, J. M. (s.f.). *Campus virtual E.T.S.I. Informatica UMA*. Obtenido de https://informatica.cv.uma.es/pluginfile.php/294380/mod_resource/content/0/apuntes/tema_3._analisis_de_requisitos.pdf
Wikipedia. (s.f.). *Wikipedia*. Obtenido de <https://es.wikipedia.org/wiki>

Apéndice A

Manual de Instalación

A.1 Requisitos

Dado que se trata de una aplicación android, el principal requisito es disponer de un dispositivo android, mas concretamente con android 5 o superior, ya que es la version mínima de android que soporta la aplicacion. Una vez cumplamos dicho requisitos, la aplicación se instalaría de la misma forma que se instalaría cualquier otra aplicación android.

A.2 Instalación de la aplicación

Podemos instalar la aplicación de múltiples formas, la más común es mediante el fichero APK de dicha aplicación, otra forma es mediante la herramienta ADB. Una vez instalada la aplicación por primera vez, esta se actualizará automáticamente cuando sea necesario.

A.2.1 Mediante la APK

Si es la primera vez que intentamos instalar una aplicación externa a la tienda de Google, lo primero que necesitaremos es habilitar la instalación de aplicaciones de orígenes desconocidos

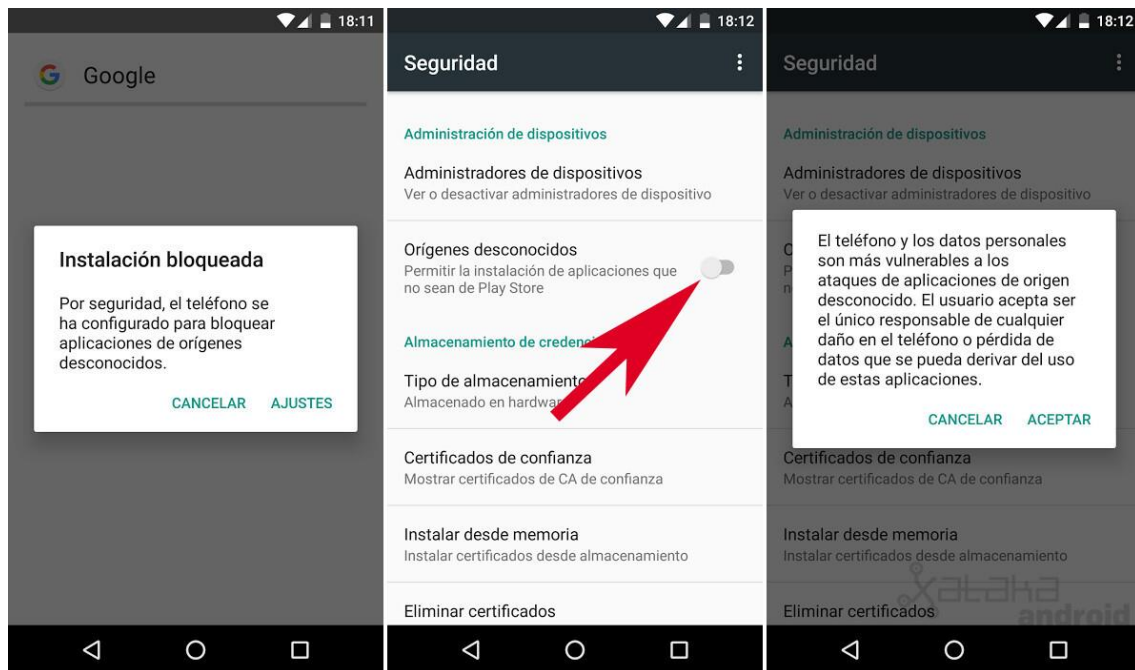


Figura 63 Pantalla habilitar orígenes desconocidos

Una vez habilitada esta opción, tendremos que tener el APK en nuestro dispositivo y buscarlo mediante cualquier administrador de archivos

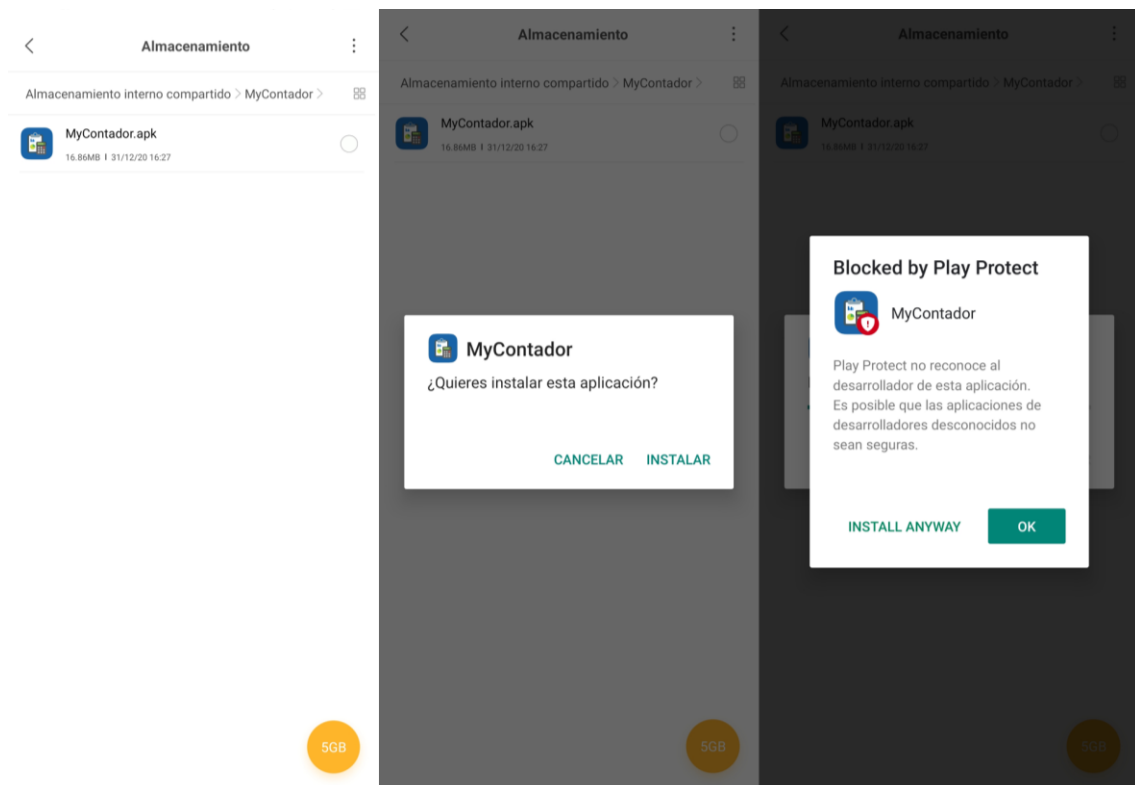


Figura 64 Pantalla de instalación de la aplicación

Una vez tengamos el APK en el dispositivo, pulsamos sobre el y le damos a instalar, si nos saliese algún mensaje preguntado si estamos seguros le damos a que si.

Una vez realizado estos pasos ya tendremos instalada la aplicación en nuestro dispositivo.

A.2.2 Mediante el proyecto

Otra opción para instalar la aplicación, es mediante Android Studio con la herramienta ADB, para ello necesitaremos un pc.

Antes de empezar, necesitaremos habilitar las opciones de desarrollador en nuestro teléfono, una vez activado, habilitaremos el modo depuración usb en los ajustes del modo desarrollador.

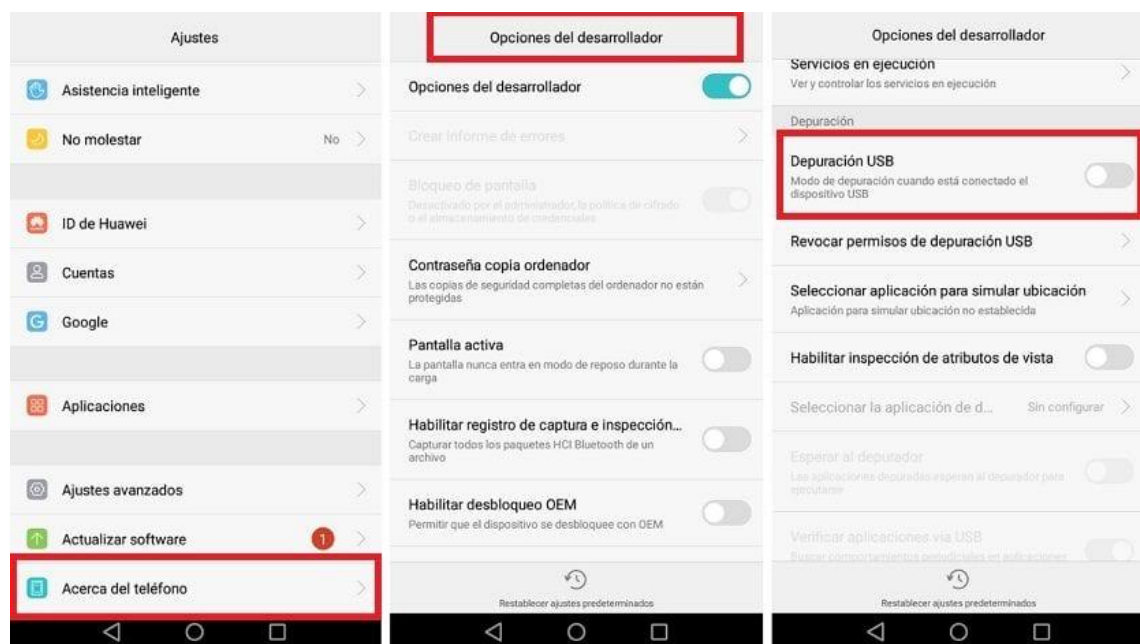


Figura 65 Habilitar modo desarrollador

En el pc lo primero será importar el proyecto mediante Android estudio.

Una vez tengamos el proyecto importado, conectamos el dispositivo android al pc mediante un cable USB, lo seleccionamos en el menú "dispositivos disponibles" y pulsamos sobre el botón "Ejecutar aplicación"

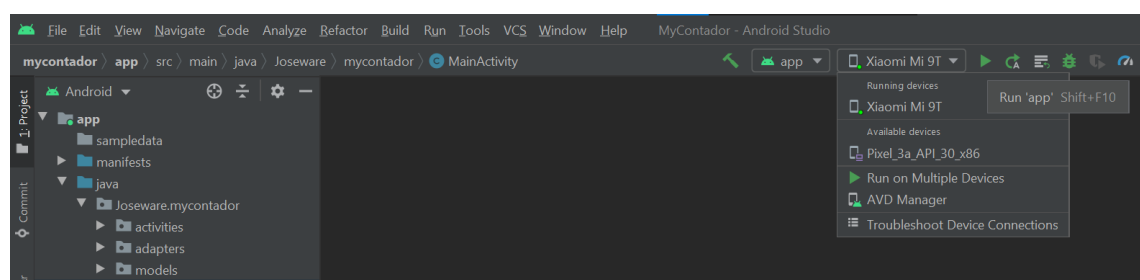


Figura 66 Ejecución de la aplicación en Android Studio

A.3 Instalación de la base de datos

Para que nuestra aplicación funcione necesitamos conectarla a un proyecto en Firebase, para ello necesitamos crear una cuenta de Google si aún no tenemos.

Una vez tengamos una cuenta de Google el siguiente paso es iniciar sesión en Firebase console y crear un nuevo proyecto.

Al crear el nuevo proyecto se nos pedirá un nombre el cual podremos poner el que queramos.

Una vez tengamos el proyecto creado, tenemos que agregar la aplicación Android, en el apartado "Nombre del paquete" introducimos el id de nuestra aplicación: "Joseware.mycontador". Los demás campos podemos dejarlos en blanco.

Una vez realizado esto, firebase nos proporcionara un fichero JSON el cual tendremos que colocar en la carpeta raíz de nuestro proyecto Android.

También tendremos que irnos al apartado "Cloud firestore" donde tendremos que permitir el acceso a la base de datos y tambien a "authentication" para permitir el inicio de sesión de usuarios.

Una vez tengamos nuestro proyecto enlazado con el servidor de firebase, nuestra app se encargara de crear la estructura de la base de datos, para así crear los campos básicos.

Una vez tengamos la estructura de la base de datos, podremos protegerla indicando las siguientes rules:

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {

    match /users/{userId} {
      allow read, update, delete: if request.auth.uid == userId;
      allow create: if request.auth.uid != null;
    }

    match /communities/{community=**} {
      allow create, read, update, delete: if exists(/databases/{database}/documents/users/{request.auth.uid})
    }

    match /values/communities {
      allow create, read, update, delete: if exists(/databases/{database}/documents/users/{request.auth.uid})
    }

    match /reports/{report} {
      allow create: if exists(/databases/{database}/documents/users/{request.auth.uid})
    }

    match /values/updates {
      allow read;
    }
  }
}
```

Figura 67 Rules Cloud Firestore para MyContador

Tras esto, ya tendremos lista la conexión entre la app y la base de datos.

Apéndice B

Manual de Usuario

B.1 Introducción

La app MyContador es una aplicación destinada a la recogida de datos de los distintos contadores para poder tener un registro de los mismos. Esta aplicación está destinada tanto a los usuarios de dichos contadores, como a los administradores de las comunidades de usuarios.

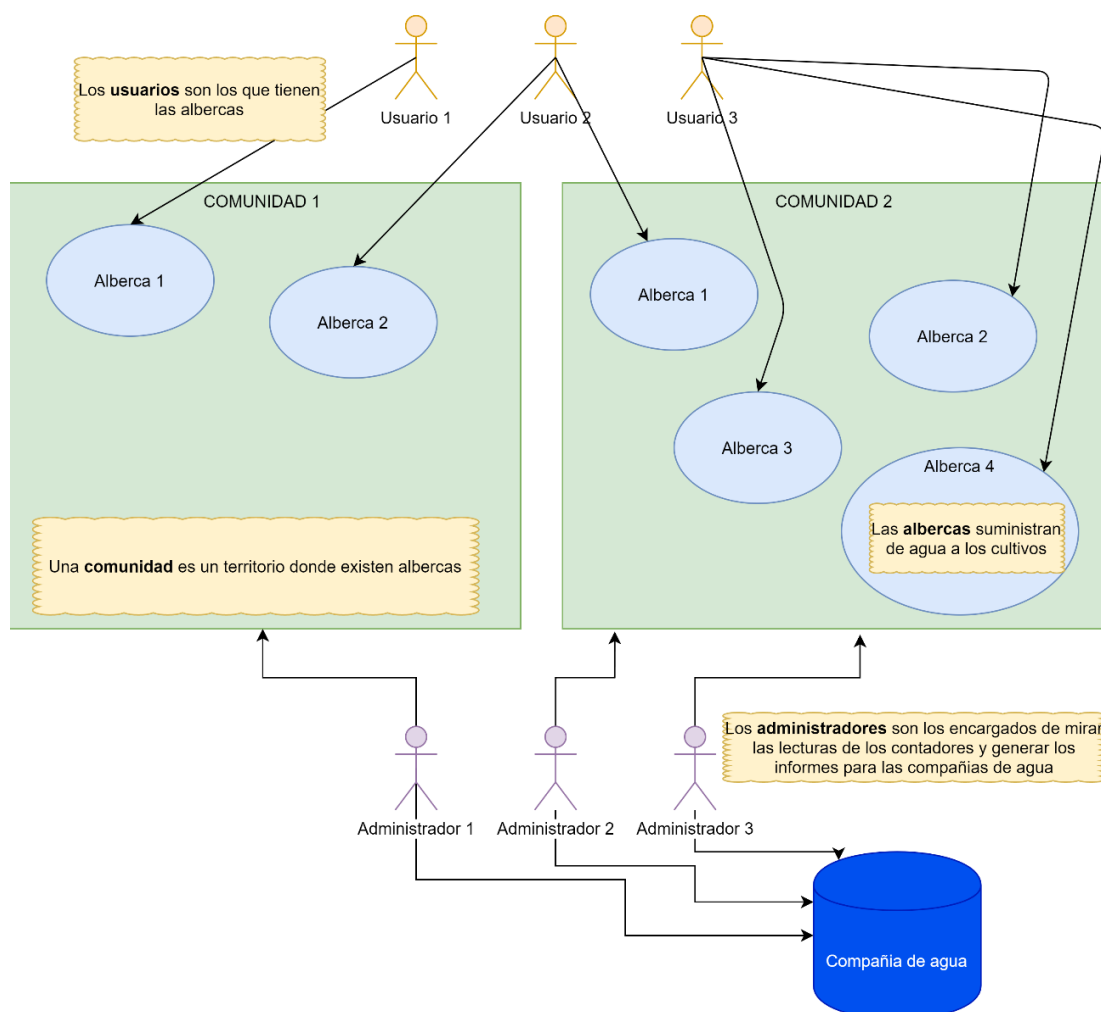


Figura 68 Descripción caso de uso MyContador

Si eres el propietario de uno o varios contadores, entonces tu perfil es el de **usuario**.
Si por el contrario eres el administrador de una comunidad y necesitas ir recogiendo y actualizando datos de distintos contadores, entonces tu perfil es el del **administrador**.

La aplicación se encarga de que los usuarios conozcan los datos sobre su contador y los administradores tengan una gestión más fácil de los contadores.



Figura 69 Interacción entre usuario y administrador

Para iniciar sesión en la aplicación lo primero es tener una cuenta de Google, si no cuentas con una, debes registrarte. Una vez tengas una cuenta de Google en tu dispositivo, inicia la aplicación MyContador.

En la pantalla principal tendrás un botón para iniciar sesión con tu cuenta de Google, selecciona la cuenta que desees usar para iniciar sesión.

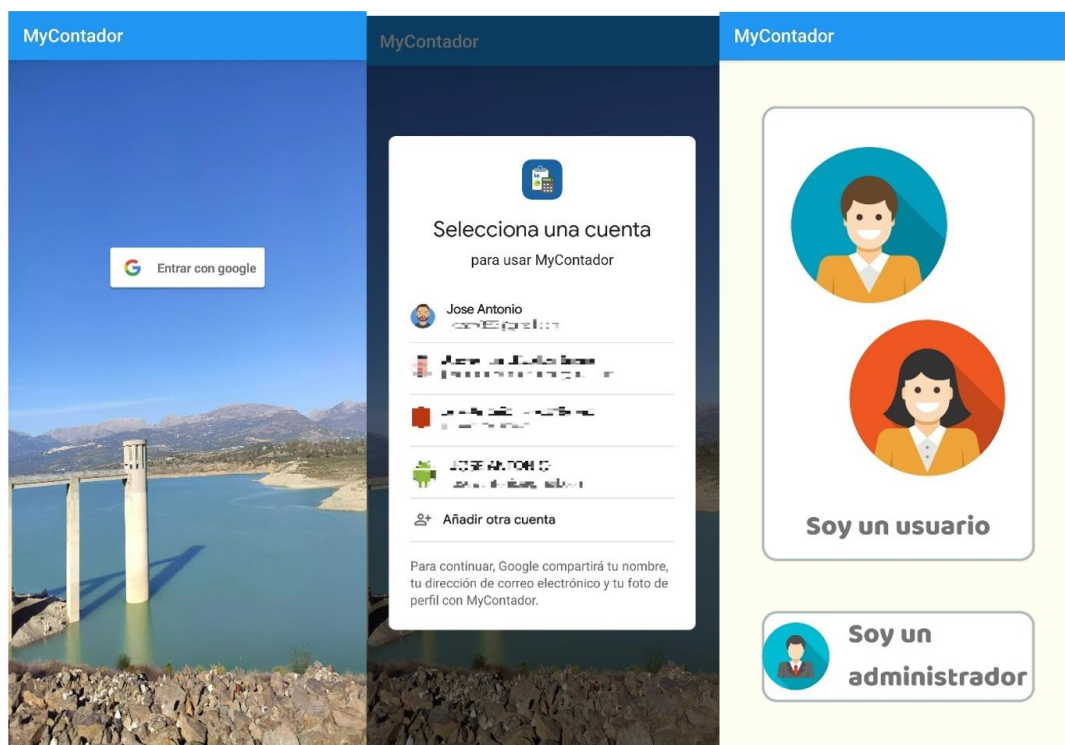


Figura 70 Inicio de sesión en la app

Una vez hayas iniciado sesión, podrás seleccionar si deseas acceder como un usuario o como administrador.

B.2 Usuario

Si accedes como usuario, recuerda que cuando quieras usar la aplicación como administrador solo tienes que cerrar sesión y volver a entrar con la misma cuenta para acceder a la ventana de selección del tipo de usuario.

Una vez hayas iniciado como usuario, podrás realizar distintas acciones.

B.2.1 Registro del usuario

Para registrarte como usuario tan solo debes iniciar sesión con tu cuenta de google, una vez hayas iniciado sesión, pulsa sobre el botón "soy un usuario".

Si es la primera vez que inicias la aplicación, tendrás una lista vacía de contadores, para por lo que tu siguiente paso será añadir un nuevo contador.

B.2.2 Agregar un contador

Si es la primera vez que iniciamos sesión, tendremos una lista vacía de contadores, para poder añadir nuestro contador pulsamos sobre el menú y seleccionamos la opción "añadir contador"

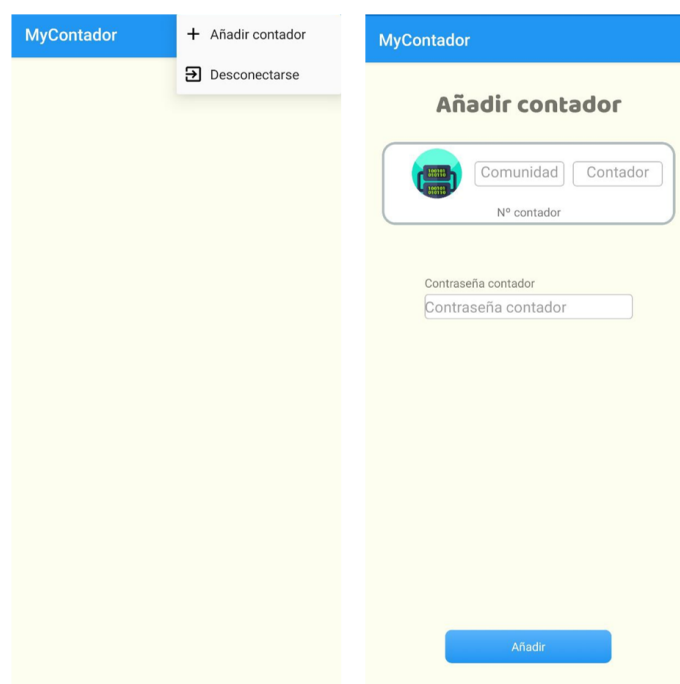


Figura 71 Añadir un contador

En la ventana de añadir contador tenemos tres campos:

- **Comunidad:** Es el numero de la comunidad a la que pertenece el contador
- **Contador:** Es el id asociado al contador automáticamente
- **Contraseña:** Es la contraseña que el administrador le ha asignado a dicho contador

Los 3 campos anteriormente mencionados solo pueden saberse preguntando al administrador, por lo que debes comunicarte con el administrador de tu comunidad para pedirle dichos datos y añadir el contador.

B.2.3 Ver los datos de un contador

Cuando visualizamos un contador, podemos ver los distintos datos asociados a este, el primer dato es su nombre, debajo de este podemos ver una barra que indica lo que hemos consumido en m³ y el porcentaje, este porcentaje depende de distintos datos como los metros cuadrados de finca y la asignación del litros a esta.

Debajo podemos observar distintos iconos con datos asociados, para ver la descripción de cada dato, tan solo tenemos que hacer clic en dicho dato y se nos mostrara un mensaje con su significado.

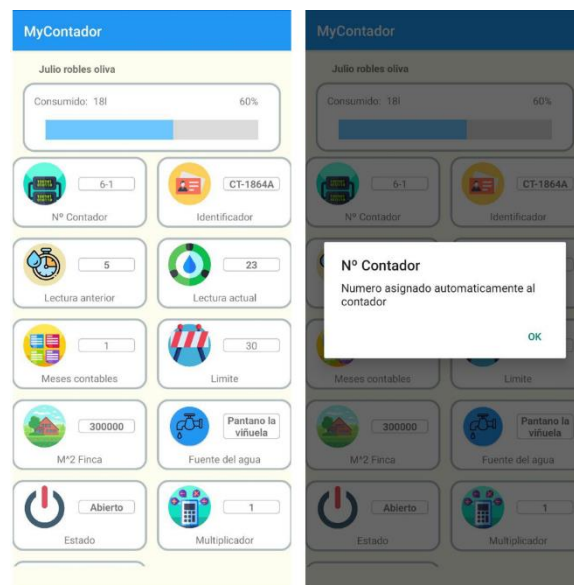


Figura 72 Datos de un contador

B.2.4 Eliminar un contador

Para eliminar un contador tenemos que irnos a la lista de contadores, una vez ahí, dejamos pulsado el contador que queremos eliminar con lo que se nos mostrara un mensaje preguntándonos si deseamos eliminarlo, pulsamos en eliminar y se eliminara el contador de nuestra lista.

Recuerda que al estar el modo usuario, el contador se eliminara de nuestra lista pero el administrador seguirá teniendo acceso a el y podrá seguir administrándolo. Si en algún momento queremos volver a tenerlo, tan solo tenemos que añadirlo de nuevo.

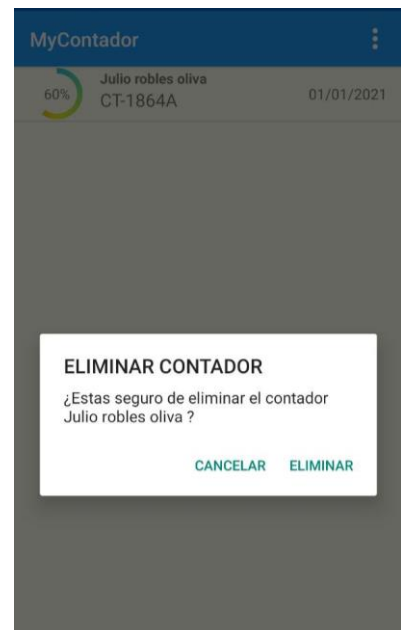


Figura 73 Eliminar un contador

B.2.5 Notificaciones de la aplicación

Los contadores pueden recibir dos tipos de notificaciones, las cuales pueden ser:

- Notificaciones de comunidad: Son las notificaciones que se envían a todos los contadores de una comunidad.
- Notificaciones del administrador: Son las notificaciones individuales que envía un administrador a un único contador.

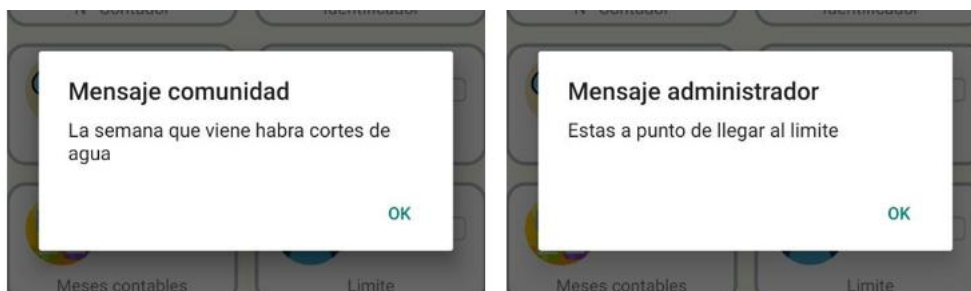


Figura 74 Mensajes del administrador

B.3 Administrador

Si accedes como administrador, recuerda que cuando quieras usar la aplicación como usuario solo tienes que cerrar sesión y volver a entrar con la misma cuenta para acceder a la ventana de selección del tipo de usuario.

Una vez hayas iniciado como administrador, podrá realizar distintas acciones.

B.3.1 Registro del administrador

Para registrarte como administrador tan solo debes iniciar sesión con tu cuenta de google, una vez hayas iniciado sesión, pulsa sobre el botón "soy un administrador".

Si es la primera vez que inicias la aplicación, tu primer paso será crear o unirse a una comunidad, ya que es esta la que agrupa los contadores. Una vez administres una comunidad, podrás administrar los contadores asociados a esta.

B.3.2 Crear una comunidad

Para crear una comunidad pulsaremos sobre el menú y seleccionamos la opción "comunidades", volvemos a pulsar el menú y pulsamos sobre "crear comunidad". Una vez estemos en esta ventana, indicamos el nombre y la contraseña. La contraseña se generará automáticamente si no la indicamos, por el contrario el nombre es obligatorio. Una vez tengamos los campos rellenos pulsamos sobre crear y se creará la nueva comunidad.

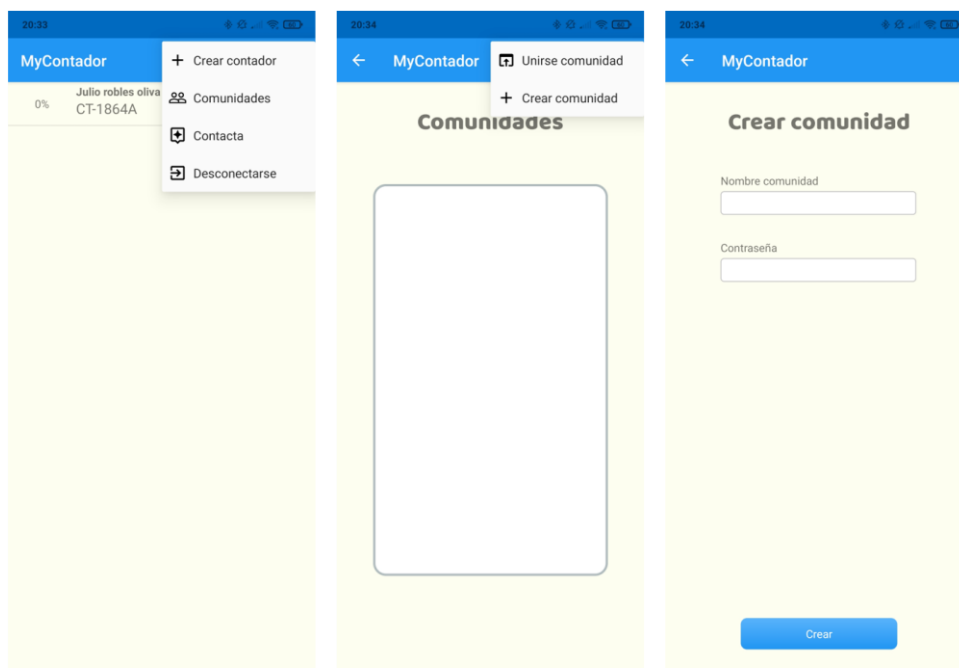


Figura 75 Creación de una comunidad

B.3.3 Unirse a una comunidad

Para unirnos a una comunidad pulsaremos sobre el menú y seleccionamos la opción "comunidades", volvemos a pulsar el menú y pulsamos sobre "Unirse comunidad". Una vez estemos en esta ventana, indicamos el nombre y la contraseña. Una vez tengamos los campos rellenos pulsamos sobre crear y se creará la nueva comunidad.

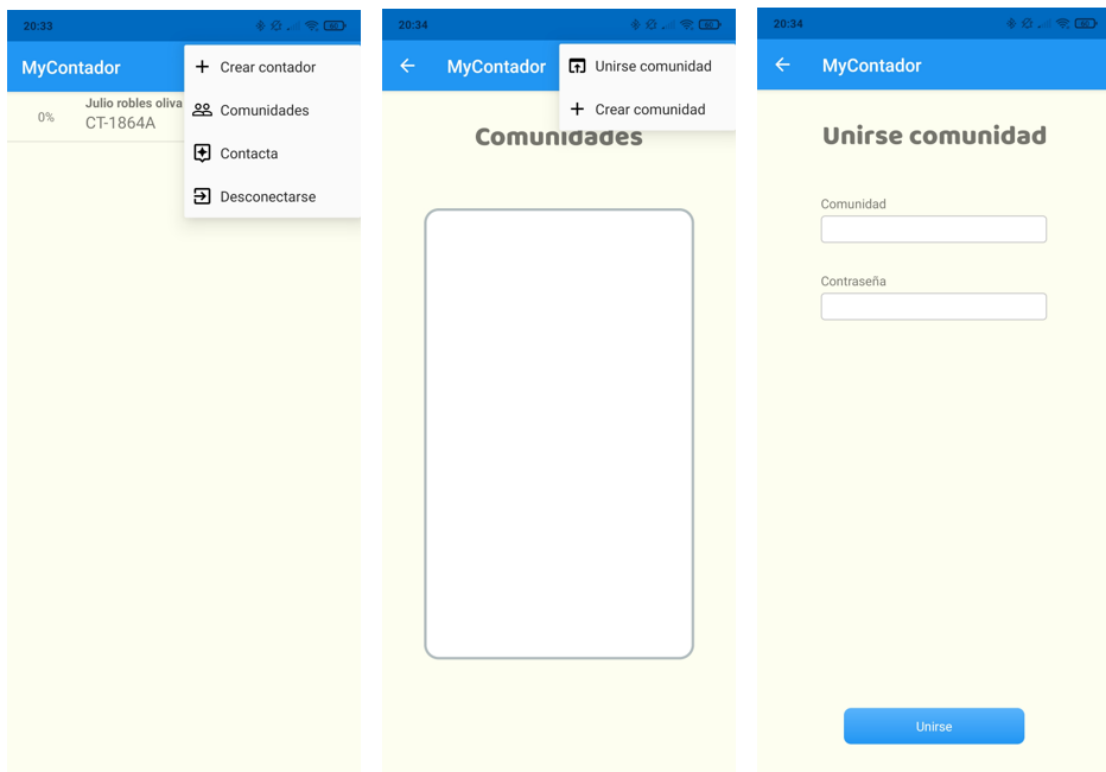


Figura 76 Pasos para unirse a una comunidad

B.3.4 Eliminar una comunidad

Cuidado cuando quieras eliminar una comunidad, ya que al eliminar una comunidad se eliminarán todos los contadores asociados a esta sin posibilidad de recuperarlos.

Para eliminar una comunidad nos iremos a la ventana de la lista de comunidades y dejamos pulsada la comunidad, una vez ahí, dejamos pulsada la comunidad unos segundos, se nos aparecerá un mensaje preguntando si queremos eliminar la comunidad y pulsamos sobre el botón eliminar.



Figura 77 Mensaje de borrar comunidad

B.3.4 Abandonar una comunidad



Figura 78 Mensaje de abandonar comunidad

Si eres el administrador de una comunidad pero no has creado dicha comunidad, no podrás eliminar la comunidad, aunque si que podrás abandonarla cuando desees, para abandonarla debes seguir los mismos pasos que para eliminarla, la aplicación detectará automáticamente que no eres el creador y te mostrará un mensaje de abandonar la comunidad.

Para abandonar la comunidad mantén pulsada la comunidad en la lista de comunidades y selecciona eliminar en el mensaje.

B.3.5 Crear un contador

Para crear con contador debemos ir a la lista de contadores, desplegar el menú y pulsar sobre "crear contador".

El único campo obligatorio que deberemos rellenar es el identificador del contador, pudiendo dejar el resto de campos en blanco para introducirlos mas tarde. Una vez tengamos rellenos los campos obligatorios pulsaremos sobre "Crear contador" para añadir nuestro contador

Figura 79 Ventana de creación de un contador

B.3.6 Eliminar un contador

Para eliminar un contador nos iremos a la lista de contadores y dejaremos pulsado durante unos segundos el contador que queremos eliminar, tras esto, nos aparecerá un mensaje preguntándonos si estamos seguros de eliminar el contador, pulsamos sobre "eliminar" para eliminar definitivamente el contador.

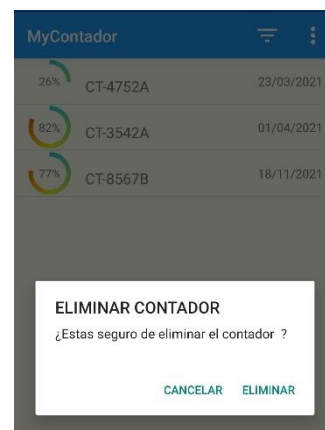


Figura 80 Mensaje de eliminación de un contador

B.3.7 Modificar valores de un contador

Los contadores tienen tres tipos de datos:

- Datos visibles por el usuario: Son los datos que el usuario puede visualizar en su pantalla. Existen dos tipos, los que el administrador puede editar, y los que no, ya que se generan automáticamente a partir de otros valores.
- Datos visibles por el administrador: Son los datos que solo el administrador puede visualizar sobre un contador.
- Acciones sobre el contador: Son las distintas acciones que pueden activarse o desactivarse sobre un contador.

Para modificar los valores de un dato pulsamos sobre él, apareciendo una ventana con la descripción del dato y el valor, introducimos el valor y pulsamos en guardar para actualizar el valor.

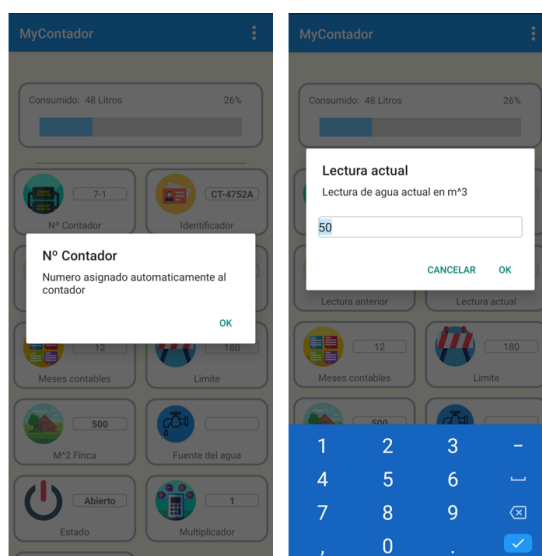


Figura 81 Modificación del valor de un contador

B.3.8 Acciones sobre un contador

Los contadores tienen una serie de valores, pero, además, también podemos activar o desactivar diferentes características como:

- Fecha visible para usuarios: Permite que los usuarios puedan ver la fecha de última lectura o se le oculte.

Estas características se sitúan como una lista al final de la vista del contador.

B.3.9 Ajustes de una comunidad

Cuando creamos una comunidad esta se crea con un usuario y una contraseña, si en algún momento deseamos cambiar dichos valores, tenemos que seleccionar dicha comunidad en la lista de comunidades, pulsamos sobre ajustes, editamos el parámetro que deseamos y pulsamos sobre "guardar". Tras esto actualizaremos el nombre y contraseña introducidos anteriormente.



Figura 82 Ajustes de una comunidad

B.3.9 Los filtros

La aplicación cuenta con una serie de filtros para mostrar los contadores que cumplan ciertas características, los filtros que tenemos son:

- Nombre: Muestra los contadores que contengan la cadena introducida como nombre.
- Comunidad: Muestra los contadores que contengan el número de la comunidad, el cual se genera automáticamente al crearse y puede visualizarse en la lista de comunidades o en los datos de cualquier contador perteneciente a dicha comunidad.
- Fuente: Muestra los contadores que contengan el origen de la fuente de agua indicado, indicada por el administrador.

- Mensaje privado: Muestra los contadores que contienen un mensaje privado si se deja en blanco o los que contengan dicho mensajes si se introduce algún valor.
- Estado: Muestra los contadores que tengan el estado indicado, como por ejemplo abierto, cerrado, en mantenimiento, etc.
- Fecha: Muestra los contadores que tengan la fecha de ultima lectura dentro del intervalo.
- Porcentaje: Muestra los contadores cuyo porcentaje este dentro del margen.

Tambien podemos leer la descripción de dichos filtros pulsando sobre el icono de cada uno de ellos.

Para acceder a dichos filtros, pulsamos sobre el botón de filtros en la ventana de contadores. Se permite el filtrado con múltiples filtros simultáneos, para ello hay que seleccionar los filtros deseados y pulsar sobre el botón filtrar o volver a la pantalla anterior, tras esto, el icono del filtro cambiará de color y se nos mostrará una lista con los filtros seleccionados.

Los filtros se guardan en el servidor de la aplicación, por lo que podrás iniciar sesión desde otro dispositivo y conservar los filtros que tenias anteriormente.

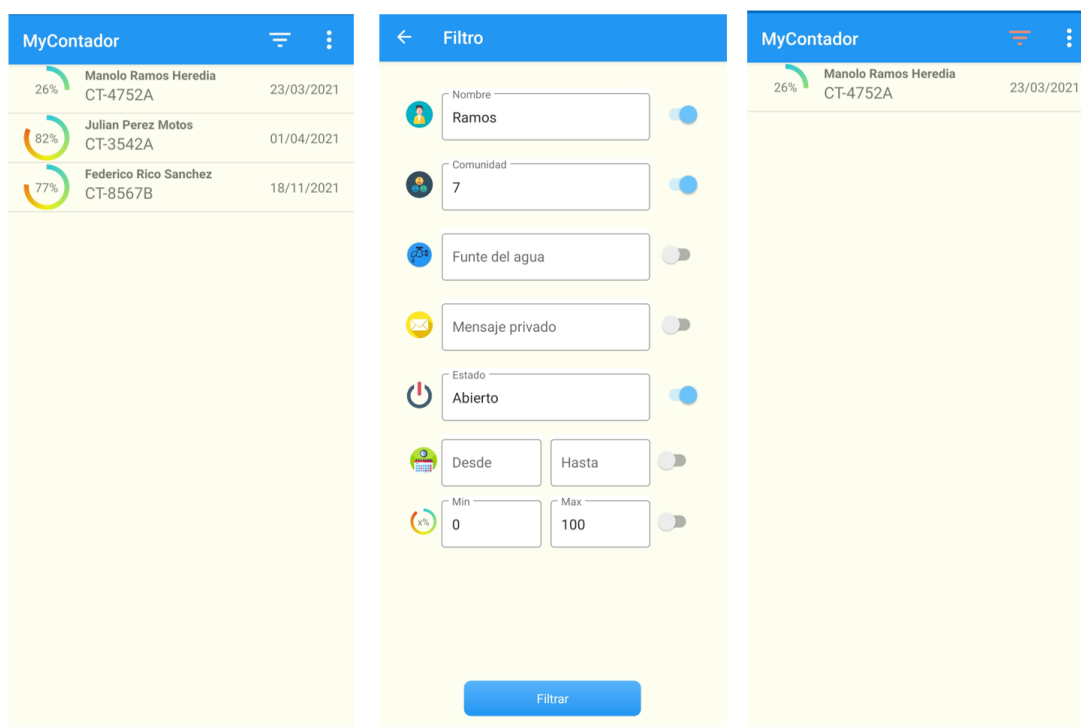


Figura 83 Filtros de la aplicación

B.3.9 Acciones sobre la lista de contadores

Una comunidad ofrece una serie de acciones, las cuales se ejecutan a la vez sobre todos los contadores pertenecientes a dicha comunidad, para ello, selecciona la comunidad que desees y pulsa el botón de la acción que quieras realizar, una vez lo pulses, se abrirá una ventana con la descripción de la acción a realizar sobre todos los contadores y un botón de confirmación . Las acciones disponibles son las siguientes.



Figura 84 Acciones de una comunidad

B.3.9.1 Borrar fechas

Borra la fecha de ultima lectura de todos los contadores, pasando esta fecha a estar en blanco.

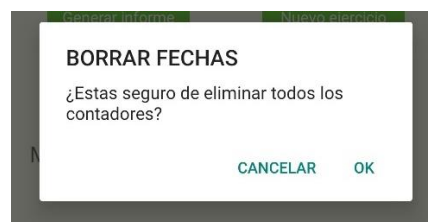


Figura 85 mensaje de borrado de fechas

B.3.9.2 Borrar mensajes

Borra todos los mensajes privados que se hayan escrito en todos los contadores, por lo que no habrá mensajes que salgan cuando un usuario visualiza un contador si este tenia algún mensaje de notificación.

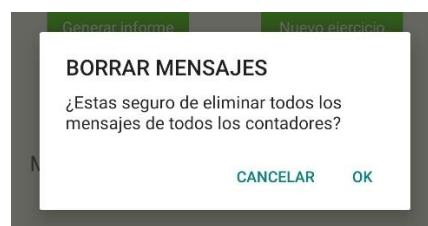


Figura 86 Mensaje de borrado de mensajes

B.3.9.3 Nuevo ejercicio

Pone el valor de la ultima lectura en la penúltima lectura, haciendo que todos los contadores pasen a haber consumido 0.

Esta función es útil cuando ya que ha medido todos los contadores y se desea empezar de nuevo a medir contadores, reiniciando lo consumido para todos estos.

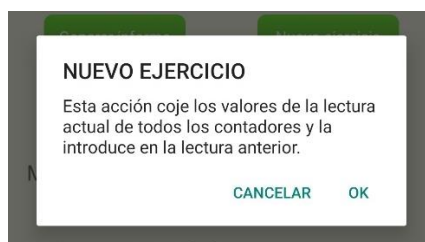


Figura 87 Mensaje de nuevo ejercicio

B.3.10 Generación de los informes

Con la generación de informes podemos exportar todos los datos de los contadores de una comunidad para generar un fichero Excel con todos ellos.

Para ellos seleccionamos la comunidad, y pulsamos sobre "Generar informe".

Una vez estemos en esta pantalla se nos mostrará todos los campos de un contador, seleccionamos los campos que queremos exportar.

Tambien podemos cambiar el orden de dichos campos, para ello dejamos pulsado el campo que queremos mover y lo desplazamos hacia arriba o abajo para cambiar su posición.

Una vez tengamos listo la selección y el orden de los campos, pulsamos sobre "Generar informe". Android nos mostrará un mensaje preguntándonos donde deseamos guardar el archivo, seleccionamos donde queremos guardarlo y se exportará.

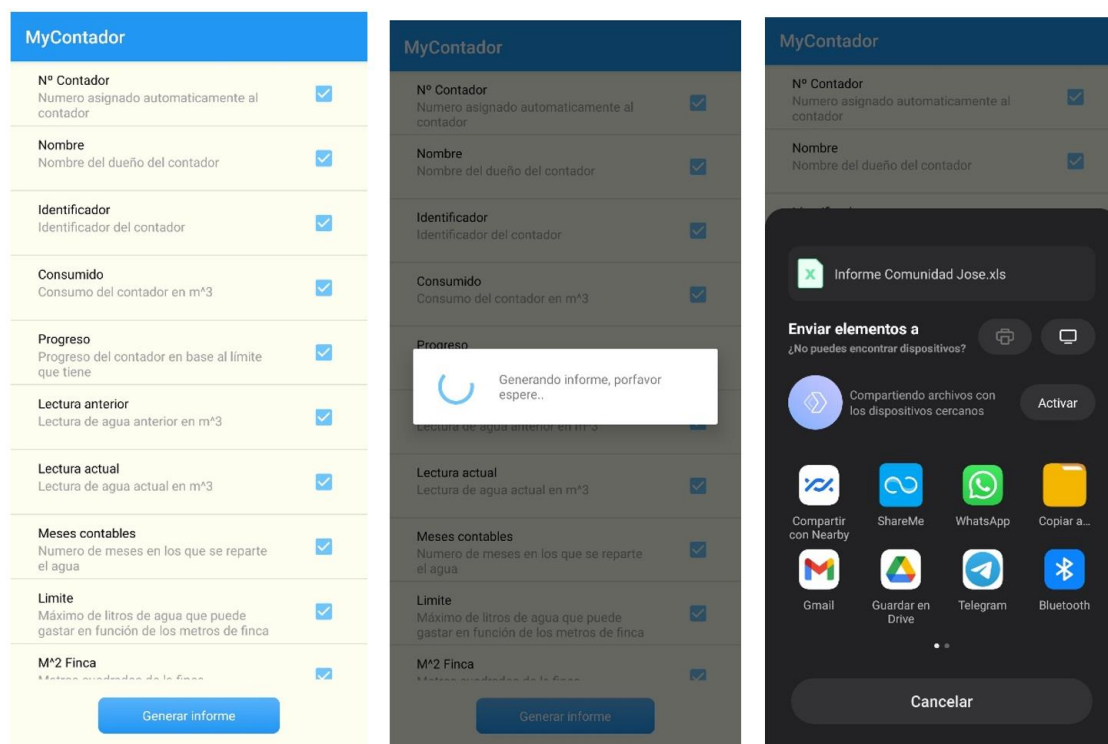


Figura 88 Pasos para la generación de informes

Una vez tengamos el fichero con los datos, tan solo necesitaremos cualquier programa que nos permita abrir un fichero Excel para visualizarlo o editarlo.

Informe Comunidad Jose.xls - Modo de compatibilidad																			
Archivo Inicio Insertar Disposición de página Fórmulas Datos Revisar Vista Ayuda																			
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	N° Contador	Nombre	Identificador	Consumido	Progreso	Lectura anterior	Lectura actual	Meses contables	Limite	M*2 Finca	Fuente del Estado	Multiplicad	Fecha ultir	Observaci	Mensaje	Telefono	Contrase	Latitud	Longitud
2	1	Manolo Ra	CT-4752A	48.0	26	2.0	50.0	12.0	180.0	500.0	Abierto	1.0	23/03/202			0.0	csjDSU	0.0	0.0
3	2	Julian Perr	CT-3542A	594.0	82	6.0	600.0	12.0	720.0	2000.0	Abierto	1.0	01/04/202			0.0	6LJ60P	0.0	0.0
4	3	Federico F	CT-8567B	84.0	77	39.0	123.0	12.0	108.0	300.0	Abierto	1.0	18/11/202			0.0	xKFnl	0.0	0.0
5	4	Ruben Vill	CT-8264B	58.0	51	0.0	68.0	12.0	216.0	500.0	Abierto	1.0	02/08/202			0.0	TX44JZ	0.0	0.0
6	5	Daniela Or	CT-9275B	16.0	7	37.0	53.0	12.0	216.0	600.0	Abierto	1.0	28/03/202			0.0	J9slbE	0.0	0.0
7	6	Alicia Lope	CT-7453B	20.0	16	584.0	604.0	12.0	124.20000	345.0	Abierto	1.0	12/11/202			0.0	qdZYZ	0.0	0.0
8	7	Julian Beni	CT-3684B	11.0	122	0.0	11.0	12.0	9.0	25.0	Abierto	1.0	02/01/202			0.0	k3LEcd	0.0	0.0
9																			
10																			
11																			

Figura 89 Informe generado en Excel

B.3.11 Estadísticas

Para poder visualizar diferentes estadísticas sobre los contadores de la comunidad debemos seleccionar la comunidad, a continuación, pulsamos sobre el botón "características", seguidamente se nos mostrará una ventana con los siguientes datos:

- Gráfica de porcentajes: Muestra una gráfica, con el número de contadores que está dentro de los porcentajes indicados. Esta gráfica es dinámica, ya que puede hacer zoom para visualizar mejor las estadísticas que muestra.
- Contador mayor porcentaje: Muestra el contador que tiene el mayor consumo de todos en base a su límite.
- Contador menor porcentaje: Muestra el contador que tiene el menor consumo de todos en base a su límite.
- Contador con mayor consumo: Muestra el contador que más ha consumido.
- Contador con menos consumo: Muestra el contador que menos ha consumido.

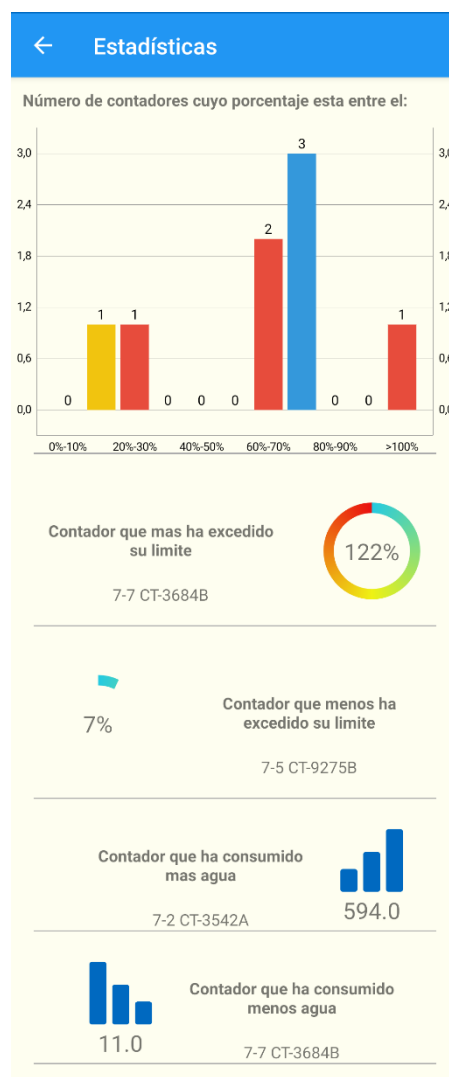


Figura 90 Estadísticas de la comunidad

B.3.12 Valores de la comunidad

Todos los contadores de una comunidad necesitan de unos valores que son los mismos para todos ellos, por ello, en vez de añadirse en cada contador, se asocian a la comunidad, estos valores son los siguientes:

- Meses contables: Es el número de meses entre los que se reparte el agua para cada contador.
- M³/Mes concedidos: Es el número de metros cúbicos que se asignan para cada mes.
- Noticia general: Es un mensaje el cual aparecerá como notificación para todos los usuarios que visualicen un contador que pertenezca a dicha comunidad.

Para cambiar algunos de estos valores seleccionamos la comunidad, luego pulsamos sobre el valor y se nos abrirá una ventana con la descripción del valor, introducimos el nuevo valor y pulsamos aceptar para cambiar dicho valor.

The screenshot displays a web interface for managing a community named 'Comunidad Jose'. At the top, there is a blue header with the word 'Comunidad'. Below this, the title 'Comunidad Jose' is centered. A section labeled 'Id comunidad' contains a text input field with the number '7'. To the left of this field is a blue button labeled 'Ajustes'. Below the 'Id comunidad' section, there are five green buttons arranged in two rows: 'Borrar fechas', 'Borrar mensajes', 'Generar informe', 'Nuevo ejercicio', and 'Estadísticas'. Further down, there are three sections, each with a label and a value displayed on a horizontal line: 'Meses contables' with the value '12.0', 'M^3/Mes Concedidos' with the value '300.0', and 'Noticia general'.

Figura 91 Valores de una comunidad



UNIVERSIDAD
DE MÁLAGA

| **uma.es**

E.T.S. DE INGENIERÍA INFORMÁTICA

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga